Final Report

W15QKN-05-D-0011, Task Order 43

(September 15, 2009)

Submitted by S. Tewksbury

Contract Number: W15QKN-05-D-0011, Task Order 43

Contract Name: Embedded Intelligent Sensor Network Systems

Task 2

Heterogeneous Multi-Robot Multi-Sensor Platform for Intruder Detection

Subtask 1
Robot/sensor localization and tracking

Prof. Yingying Chen
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining th data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducin this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a curren valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**		
1. REPORT DATE *(DD-MM-YYYY)* 09-15-2009	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From - To)* 08-30-2008 – 09-30-2009
4. TITLE AND SUBTITLE ***Heterogeneous Multi-Robot Multi-Sensor Platform for Intruder Detection***		**5a. CONTRACT NUMBER** **W15QKN-05-D-0011**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) *Prof. Yingying Chen*		**5d. PROJECT NUMBER**
		5e. TASK NUMBER 43
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) *Department of Electrical and* *Computer Engineering* *Stevens Institute of Technology* *Hoboken, NJ 07030*		**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) DoD-ARDEC ACOE Building 407, Picatinny 07806		**10. SPONSOR/MONITOR'S ACRONYM(S)**
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A Unlimited Distribution		
13. SUPPLEMENTARY NOTES 3		

14. ABSTRACT

Localization of mobile sensors such as service robots in tactical mobile sensor networks is important because the location of mobile sensors is a critical input to many higher-level tasks, such as intruder detection, tracking, monitoring and geometric based protection. In this year's work, we focused on the following research tasks: (1) we developed and deployed a prototype of the localization system in the ECE Dept. at Stevens for both 802.11 (WiFi) networks as well as 802.15.4 (ZigBee) networks to localize the mobile sensor nodes; (2) we investigated the performance of wireless localization using signal strength on commodity hardware embedded in mobile robots. Our work relies on trace-driven analysis using an extensive experimental infrastructure; (3) we developed a technique to detect the co-moving transmitters through similarities of the received signals; and (4) we investigated to perform initial intrusion detection

ZigBee 802.15.4,Wireless Localization, Ropbot Patrolling, RFID, Time of Arrival Received Signal Strength, Non-Linear Least Squares

16. SECURITY CLASSIFICATION OF: **UNCLASSIFIED**			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSO** Shafik Quoraishee
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	SAR	51	**19b. TELEPHONE NUMBER** *(include are code)* 9737249462

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Abstract

Localization of mobile sensors such as service robots in tactical mobile sensor networks is important because the location of mobile sensors is a critical input to many higher-level tasks, such as intruder detection, tracking, monitoring and geometric based protection. In this year's work, we focused on the following research tasks: (1) we developed and deployed a prototype of the localization system in the ECE Dept. at Stevens for both 802.11 (WiFi) networks as well as 802.15.4 (ZigBee) networks to localize the mobile sensor nodes; (2) we investigated the performance of wireless localization using signal strength on commodity hardware embedded in mobile robots. Our work relies on trace-driven analysis using an extensive experimental infrastructure; (3) we developed a technique to detect the co-moving transmitters through similarities of the received signals; and (4) we investigated to perform initial intrusion detection using signal variations.

1. Introduction

Technology trends have reduced the cost of wireless networking to the point where it can be added to nearly every computing device, such as the mobile robot. Indeed, wireless networking devices include laptops, PDAs, small sensors, active RFID tags, and even cameras and printers. The inclusion of wireless networking in such a broad range of devices opens an opportunity for a new computing service: positioning devices in physical space. A generic service of this kind would enable a host of additional applications, ranging from such diverse areas as asset management, disaster recovery, inventory tracking, geometry-based routing, and perimeter-based security. Using the same wireless traffic for both communication and positioning would provide tremendous cost and deployment savings over an independent localization infrastructure.

We developed and deployed a prototype of the localization system in the ECE Dept. at Stevens for both 802.11 (WiFi) networks as well as 802.15.4 (ZigBee) networks to localize the mobile sensor nodes. Our localization system works with anchor-based approach. The service robots and sensors will be localized by a localization server, which responses to report the position information periodically to the service robots.

Recent years have witnessed the development of a plethora of localization techniques. Compared to various physical properties of radio signal, such as Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), using the Received Signal Strength (RSS) [1]–[3] is an attractive approach to perform localization since it can reuse the existing wireless infrastructure and presents a tremendous cost savings over deploying localization-specific hardware. Therefore, we investigated the performance of wireless localization using signal strength on commodity hardware embedded in mobile robots. Our work relies on trace-driven analysis using an extensive experimental infrastructure based on our deployed localization system.

Many location-aware applications benefit from higher level information about the movements of robots and sensors. One instance of such higher-level information is co-movement, which describes whether a set of mobile sensors are moving together on a common path. Co-movement information could be used to infer containment relationships and help to track multiple mobile sensors. In our work, we conducted initial investigation of detecting co-movement through correlated signal variations over time rather than directly measuring the signal difference between two transmitters. Moreover, we exploit Received Signal Strength (RSS) obtained from the existing wireless infrastructures for performing intrusion detection when the intruders or objects do not have any radio devices attached to them.

2. Approach Taken

2.1 Task 1: Localization System Prototype

Our localization system prototype is designed with fully distributed functionality and easy to plug-in localization algorithms [4]. It is built around 4 logical components: Transmitter (robots or sensors), Landmark, Server, and Solver. The system architecture is shown in Figure 1.

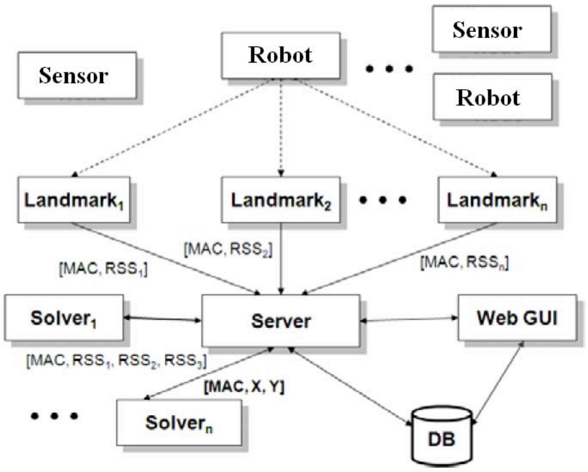


Figure 1: Localization testbed system components.

Robot: Any robots equipped with RF device can be localized. Often the application code does not need to be altered on a sensor node and robot in order to localize it.

Landmark (Anchor): The Landmark component listens to the packet traffic and extracts the RSS reading for each robot or sensor. It then forwards the RSS information to the Server component. The Landmark component is stateless and is usually deployed on each landmark or access point with known locations.

Server: A centralized server collects RSS information from all the Landmark components. The spoofing detection is performed at the Server component. The Server summarizes the RSS information such as averaging or clustering, then forwards the information to the Solver component for localization estimation.

Solver: A Solver takes the input from the Server, performs the localization task by utilizing the localization algorithms plugged in, and returns the localization results back to the Server. There are multiple Solver instances available and each Solver can localize multiple transmitters simultaneously.

During the localization process, the following steps will take place:
1. A robot sends a packet. Some numbers of Landmarks observe the packet and record the RSS.
2. Each Landmark forwards the observed RSS from the transmitter to the Server.
3. The Server collects the complete RSS vector for the transmitter and sends the information to a Solver instance for location estimation.
4. The Solver instance performs localization and returns the coordinates of the transmitter back to the Server.

If there is a need to localize hundreds of robots or sensors at the same time, the server can perform load balancing among the different solver instances. This centralized localization solution also makes enforcing contracts and privacy policies more tractable.

2.2 Task 2: Localization Algorithms

2.2.1 Lateration Based Algorithms

Lateration-based algorithms [5-7], explicitly model the signal-to-distance effect on RSS. They estimate the position of the transmitter by measuring the distance to multiple anchors (i.e. access points). There are two phases in RSS-based lateration methods: the off line training phase and the runtime localization phase. During the off line training phase, RSS samples are collected at various known locations from multiple access points and distances are calculated from the known locations to anchor. The measured RSS readings and distances are then used to fit the signal propagation model based on the signal to distance relationship. During the runtime localization phase, there are two steps: ranging step and lateration step. In the ranging step, according to the measured online RSS from the wireless device and the fitted signal-to-distance relationship, the distances between the wireless device and multiple access points can be estimated. In the lateration step, we can estimate the location of the device according to estimated distances based on least squares methods.

Non-Linear Least Square (NLS): Given the estimated distances d_i and known positions *(x_i, y_i)* of the *i*th access points, the position *(x, y)* of the wireless node can be estimated by finding $(\hat{x}, \hat{y})$ satisfying:

$$(\hat{x}, \hat{y}) = \arg\min_{x,y} \sum_{i=1}^{N} [\sqrt{(x_i - x)^2 + (y_i - y)} - di]^2$$

where *N* is the number of access points that used to estimate the location of the wireless node. Non-linear least square can be viewed as an optimization problem where the objective is to minimize the sum of the error square.

Linear Least Square (LLS): The LLS is an approximation of NLS solution. It linearizes the NLS problem by introducing a constraint in the formulation and obtains a closed form solution of location estimation. Compared with NLS, LLS has less computational complexity. The location of the wireless device can be obtained by solving the form ***A**x* = *b* with:

$$\mathbf{A} = \begin{pmatrix} x_1 - \frac{1}{N}\sum_{i=1}^{N} x_i & y_1 - \frac{1}{N}\sum_{i=1}^{N} y_i \\ \vdots & \vdots \\ x_N - \frac{1}{N}\sum_{i=1}^{N} x_i & y_N - \frac{1}{N}\sum_{i=1}^{N} y_i \end{pmatrix}$$

$$\mathbf{b} = \frac{1}{2}\begin{pmatrix} (x_1^2 - \frac{1}{N}\sum_{i=1}^{N} x_i^2) + (y_1^2 - \frac{1}{N}\sum_{i=1}^{N} y_i^2) \\ -(d_1^2 - \frac{1}{N}\sum_{i=1}^{N} d_i^2) \\ \vdots \\ (x_N^2 - \frac{1}{N}\sum_{i=1}^{N} x_i^2) + (y_N^2 - \frac{1}{N}\sum_{i=1}^{N} y_i^2) \\ -(d_N^2 - \frac{1}{N}\sum_{i=1}^{N} d_i^2) \end{pmatrix}$$

where ***A*** is only described by the coordinates of access points, *b* is represented by the distances to the access points together with the coordinates of access points and x is the estimated location of wireless device. Thus, the estimated location $(\hat{x}, \hat{y})$ of the wireless device is given by x = $\boldsymbol{(A^TA)^{-1}A^T}b$.

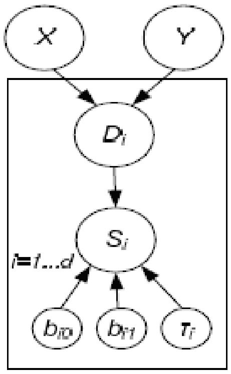


Figure 2: A simple Bayesian graphical model

Bayesian Networks (BN): BN localization is a machine learning based algorithm that infers the distribution of the coordinates of the targeted node. It uses the Bayesian Graphical Model to encode the signal-to-distance relationship for localization [8]. Figure 2 shows the basic Bayesian Network used for our study. The vertices *X* and *Y* represent a location in a two-dimensional space; the vertex s_i is the RSS reading from the *i*th access point; and the vertex D_i represents the Euclidean distance between the location specified by *X* and *Y* and the *i*th access point. The value of s_i follows the log-distance propagation model $s_i = b_{0i} + b_{1i} *logD_i$, where b_{0i}, b_{1i} are the parameters specific to the ith access point. The distance Di = p(X − x_i)2 + (Y − y_i)2 in turn depends on the location *(X, Y)* of the measured signal and the coordinates (x_i, y_i) of the *i*th access point. The network models noise and outliers by modeling the s_i as a Gaussian distribution around the above propagation model, with variance τ_i: $s_i \sim N(b_{0i} + b_{1i} *logD_i, \tau_i)$. The initial parameters (b_{0i}, b_{1i}, τ_i) of the model are unknown, and the training data is used to adjust the specific parameters of the model according to the relationships encoded in the network. In general, there is no closed form solution for the returned joint distribution of the *(X, Y)* location. We use a Markov Chain Monte Carlo (MCMC) simulation approach to draw samples from the joint density. BN returns the sampling distribution of the possible location of *X* and *Y* as the localization result.

2.2.1 Classification Based Algorithms

Classification algorithms (i.e. matching algorithms), do not rely on a model of signal strength and distance relationship. Rather, they match RSS observations against an existing signal map. The term classification, as used in the machine learning sense, implies that the goal of the classifier is to map a potentially large input space into a much smaller space of labels. In the case of localization, the labels are a set of discrete *(x, y)* locations.

RADAR: The RADAR algorithm is a classic scene-matching localization algorithm [1]. RADAR requires a signal map, which is a set of fingerprints with known *(x, y)* locations. Given a fingerprint with an unknown location, i.e., one to localize, RADAR returns the *x, y* of the closest fingerprint in the signal map to the one to localize, where "closest" is defined as the Euclidean distance of the fingerprints to each other in a *N*-dimensional "signal space" with *N* access points . That is, it views the fingerprints as points in an *N*-dimension space, where each access point forms a dimension, and returns the corresponding *x*, *y* of the closest point.

Gridded-RADAR (GR): GR is an improvisation over RADAR where measurement area is sub-divided into a regular grid and the signal map provided in the offline phase is interpolated over the entire grid. The online phase is similar to RADAR with the exception that the "closest" fingerprint in signal space is chosen from the interpolated signal map. This approach has an advantage of obtaining a much finer-grained resolution as the regions which are not covered by the signal map can also be returned as location estimates.

ABP: Area Based Probability (ABP) utilizes an Interpolated Map Grid (IMG) to interpolate the signal map to cover the entire experimental floor. The floor is then divided into a regular grid of equal sized tiles. Because direct measurement of the fingerprint for each tile is expensive and prohibitive for fine-grained tiles, we use an interpolation approach. The goal of using an IMG fitting is to derive an expected RSS fingerprint for each tile from the data set that would be similar to an observed one.

ABP returns a set of tiles bounded by a probability that the mobile device is within the returned tile set. The probability is called the confidence and it is adjustable by the user. ABP assumes the distribution of RSS for each landmark follows a Gaussian distribution with mean as the expected value of RSS reading vector ***s***. The Gaussian random variable from each access point is independent. ABP then computes the probability of the mobile device being at each tile L_i, with $i = 1...L$, on the floor using Bayes' rule:

$$P(Li\,|\,s) \;= \frac{P(s\,|\,Li) \times P(Li)}{P(s)}$$

Given that the mobile device must be at exactly one tile satisfying $\sum_{i=1}^{L} P(Li\,|\,s) = 1$, ABP normalizes the probability and returns the most likely tiles/grids up to its confidence α [9]. In order to normalize for accuracy and stability results, we select the tile with the median localization error from the tile set.

2.3 Task3: Detecting Co-Moving Wireless Devices

The environment in which wireless communication takes place affects the received signal power (or signal-to-noise ratio). The key idea underlying our technique is exploiting shadow fading, signal attenuation due to objects blocking the path of communication. Two transmitters, such as the RF device embedded in mobile robots, in close proximity will be similarly affected by surrounding buildings, furniture, or passing people. Therefore, the observed signal power from these transmitters should be correlated. This similarity in signal strength in turn should also translate to correlations in localization errors.

Our technique captures these similarities by calculating the correlation coefficient over a time-series trace of signal strength or location coordinate values. The correlation coefficient measures the strength of a linear relationship between two random variables. Thus the correlation coefficient captures similarities in the changes of two values, even if the absolute values are different. Our method uses the Pearson's product moment correlation coefficient [10], a preferred method for quantitative measures such as the RSSI traces used. For n samples each from two random variables X and Y , it is defined as

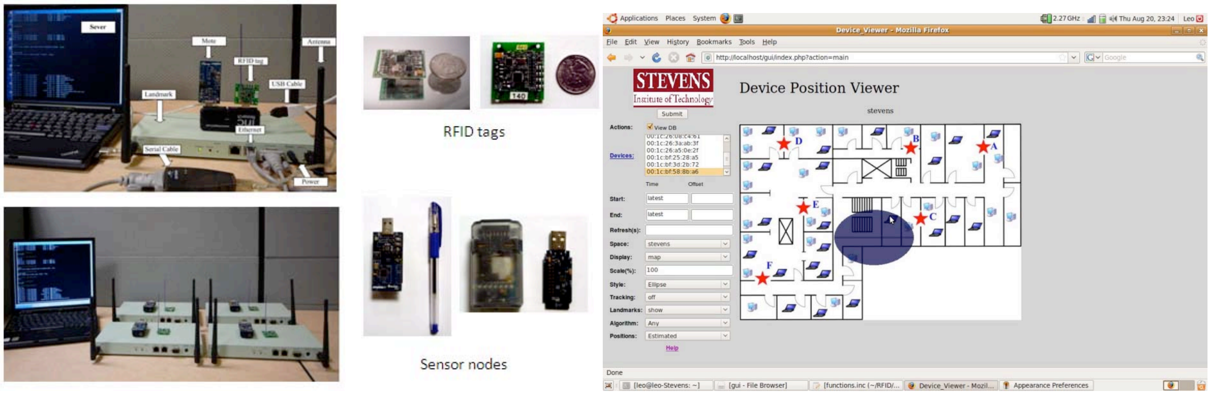
$$r_{xy} = \frac{\sum x_i y_i - n \hat{x} \hat{y}}{(n-1) S_x S_y}$$

where S_x and S_y are the sample standard deviations. The correlation coefficient lies in the interval [−1, 1], where 0 indicates no correlation, +1 indicates maximum positive correlation, and -1 indicates maximum negative correlation. We empirically determined a correlation coefficient threshold of 0.6; values that exceed this threshold indicate co-movement.

Received signal strength, however, also significantly varies due to multi-path fading. It can introduce received signal strength changes of more than 20dB between locations separated only by half the wavelength of the carrier frequency, if no line-of-sight path to the transmitter is available. These variations render the similarities due to shadow fading difficult to detect. To address this challenge, our method calculates a moving average over signals, which acts as a low-pass filter to reduce or remove multi-path effects. Movement also helps detection of shadow fading similarities, because co-moving transmitters will experience received signal strength changes due to shadowing at similar points in time (e.g., two co-moving transmitters would pass a building corner at the same time). In our prototype, we have implemented our technique by monitoring the RSSI indicators reported for each packet reception by the receiver. RSSI has been shown to be a good indicator of channel quality; hence it should provide adequate information about fading patterns. RSSI is also available across all wireless technologies, which allows measuring co-movement across different transmitters.

2.4 Task 4: Initial Intrusion Detection Using Signal Strength

Although the radio signal is affected by reflection, refraction, shadowing and scattering, the RSS at wireless devices should be relatively stable if there is no movement or changes in wireless environments. On the other hand, the wireless environment will be affected if there is a presence of intrusions, for instance, an intruder standing or walking in a wireless environment will absorb, reflect, and diffract some of the transmitted power. Consequently, the RSS at wireless devices



(a) System testbed (b) Localization results on GUI

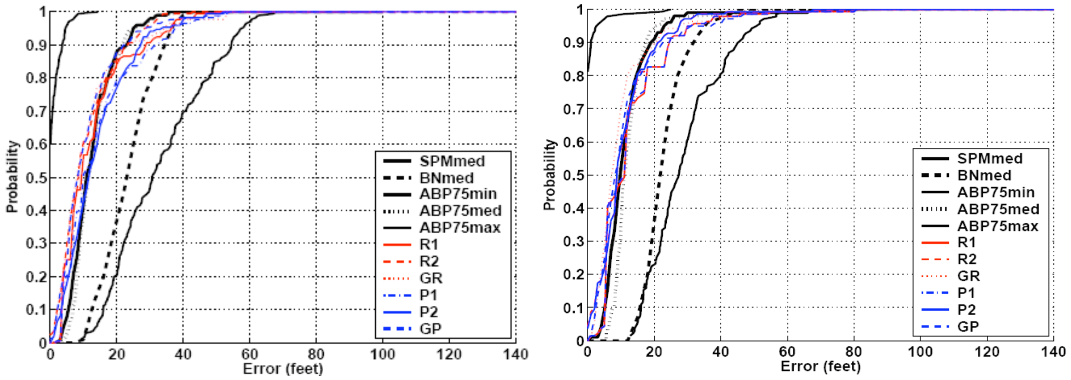
Figure 3: Localization testbed and the GUI interface of the localization system

will be impacted and results in changes of RSS values. Therefore, based on the changes of RSS at wireless devices, it is possible to detect intrusion in wireless environments.

3. Results

3.1 Localization System Prototype

A key contribution of the proposed localization system is its universal approach: it will integrate different hardware and software capabilities within a single localization framework. Moreover, we found that a centralized solution has critical advantages that are often overlooked in the literature. First, it makes cleaning and summarizing the traffic observations much easier. Second, it enables a variety of additional services, such as attack detection and tracking, to utilize the same underlying localization system. Finally, we believe that centralization makes enforcing contracts and privacy policies tractable. However, we will leave open the issues of privacy contracts and policy enforcement as future work. Figure 3 shows the localization testbed and the interface of the localization system.



(a) (b)

Figure 4: Error CDF across algorithms in two different indoor environments.

3.2 Localization Results across Algorithms

Figure 4 shows the localization performance of the algorithms for two different office buildings. For the ABP algorithms, the median tile error is presented, as well as the minimum and maximum tile errors. As in previous work, the algorithms all obtain similar performance, with the exception of BN which slightly under-performs the other algorithms.

3.3 Initial Results of Detecting Co-Moving Wireless Devices

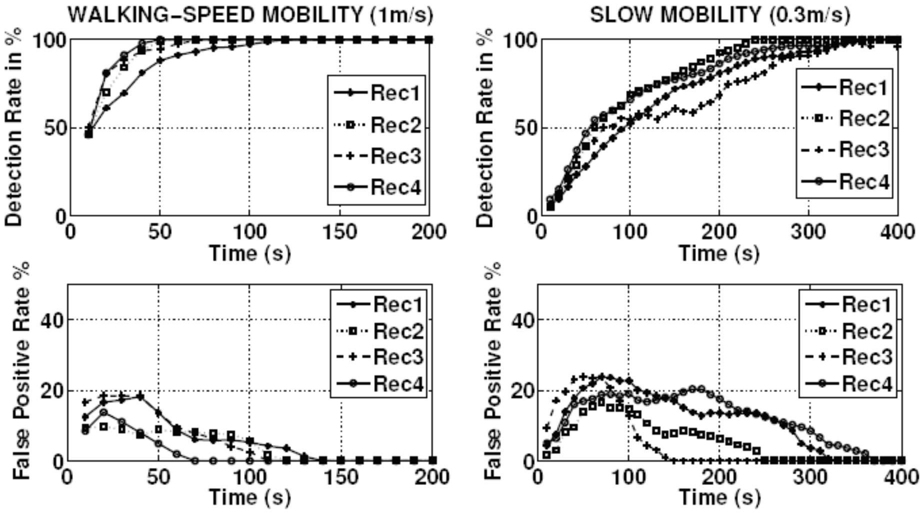


Figure 5: Effectiveness of our method in terms of detection rate and false positive rate.

To evaluate the performance of our approach, we examined the detection rate and the false positive rate of determining the co-mobile transmitters. Figure 5 depicts the detection rate and the false positive rate as a function of time with respect to each receiver for the IEEE 802.11 network for both Slow Mobility as well as Walking-Speed Mobility experiments.

Figure 5 shows that in both the Walking-Speed Mobility and Slow Mobility experiments, our technique is able to detect all co-moving and non-co-moving pairs over all the data subsets accurately. We can also see that, increasing the observation time Ts improves the co-mobility detection rate while reducing the likelihood of observing spurious matches. We found that the mobility speed also has an impact on the time required to achieve high detection rate and low false positive rate. In the Walking-Speed Mobility experiment, it takes about 130 seconds to detect all co-moving data subsets. Whereas it takes around 370 seconds to achieve the same in the Slow Mobility experiment. This suggests that, with higher speed, more shadow fading effects can be observed within a shorter duration, leading to improved detection performance. The results of the Slow Mobility experiment represent detection performance of DECODE under challenging conditions.

3.4 Initial Intruder Detection Using Received Signal Strength

Experiment scenarios: In this study, we explore two representative types of intrusion events: static and moving. We define a static event when an intruder breaks in the area of interest and moves from one position to another, at each position the intruder stands still for a certain period of time. Whereas a moving event is defined for an intruder walking or running across the area of interest. In our experiments as shown in the Figure 6, the time interval between two consecutive intrusion events in a series of events is around 180 seconds. We note that there can be multiple intruders present in the system. Since multiple intruders will cause more changes in wireless environments and have bigger impact on RSS readings, the detection of the presence of multiple intruders is easier than an individual intruder. The detailed experimental setup of each scenario and behavior of the intruder are described below.

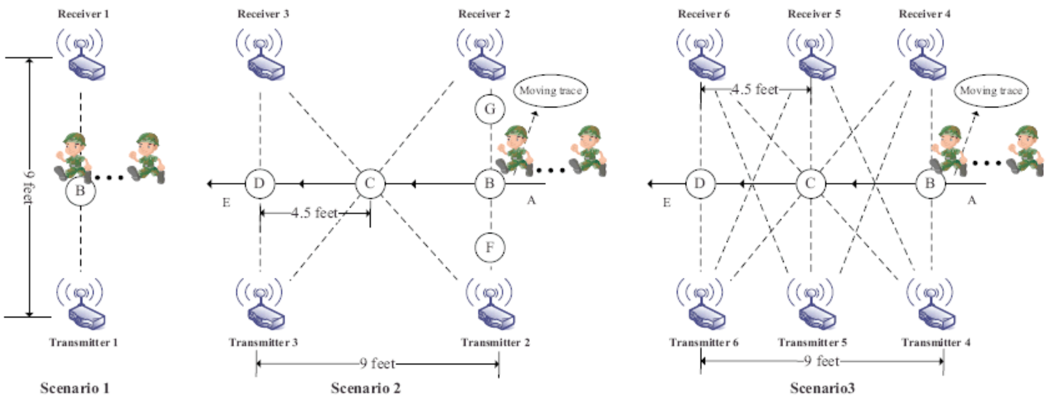
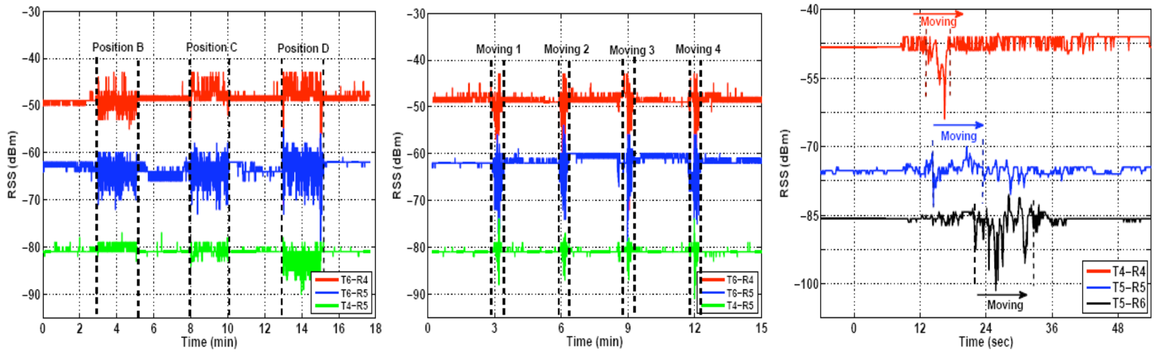


Figure 6: Experimental setup when one or more intruders are present in the system.

Experimental Scenario 1: In this scenario, there are one transmitter and one receiver in the area of interest. The distance between the transmitter and the receiver is 9 feet. This scenario may represent a low density environment in office buildings since there is just one transmitter-receiver pair which represents the wireless link between one wireless device and an access point. The receiver recorded packets for approximately 1560 seconds from the transmitter. There are three intrusion instances during this time period. For each instance, the intruder came in and stood at the center of the transmitter-receiver pair for about 120 seconds.

Experimental Scenario 2: We increased the density of the devices in this scenario, which may represent the typical density in an office building environment in which there are many wireless devices communicate with access points. There are two transmitters and two receivers deployed at four corners of the 9 feet by 9 feet square area. There are four transmitter receiver pairs in total. Two receivers recorded packets for approximately 2400 seconds from two transmitters. There are nine intrusion instances including five static cases and four moving cases during this time period. For each static intrusion instance, the intruder stood at different locations (shown as B, C, D, G and F in Figure 1) for about 120 seconds, whereas the intruder went across the experimental area for each observed moving instance.

Experimental Scenario 3: In this scenario, there are three transmitters and three receivers. The distance between two adjacent transmitter and receiver is 4.5 feet. There are nine transmitter-receiver pairs in total. The duration of this experiment is about 1800 seconds including seven intrusion instances in total with three static cases and four moving cases. The intruder stood for 120 seconds at three different positions (B, C, and D) for each static instance and went across the experiment area for each moving case. We envision there will be an increasing density of wireless devices deployed in our environments as the wireless networks become more pervasive. Thus, this set up with higher device density can help to analyze the impact of device density on diagnosing passive intrusion. In addition, wireless devices are usually not uniformly deployed. For instance, wireless devices (e.g., sensor nodes) can be deployed in a higher density in the sensitive area for asset protection and at the entrance or exit of the facility.



(a) Standing (b) Moving (c) Moving direction

Figure 7: Pattern profiling of different intrusion cases.

Pattern Profiling: By utilizing the data after filtering, we can explore various profiles to describe different intrusion patterns. In passive intrusion detection, it is essential to differentiate intrusion activities from random environmental changes. The critical property that a pattern profiling approach exhibits is that it can drive unclear or complicated situations into separate categories, which make it possible for further analysis based on categorized information. This largely helps for passive intrusion learning as we can systematically detect the intrusion and define its characteristics.

The three patterns in Figure 7 (a) represent the RSS readings for three transmitter-receiver pairs, T6-R4, T6-R5, and T4-R5 respectively when the experimenter stood at positions B, C, and D respectively. In order to examine the changes of RSS clearly, we shifted the RSS readings by 15 dBm for T6-R5 and by 30 dBm for T4-R5. We observed that there is an obvious change in RSS readings when the experimenter walked in and stood within the experimental area. Further, the results of the second experiment in Figure 7 (b) show that there is an obvious RSS pattern change for each moving instance. The key observation is that the RSS patterns when the experimenter is static are different from those when the experimenter is walking around. These results indicate that different RSS profiles can be established to distinguish the moving patterns of intruders.

Moving Direction: When the intruder is moving around, determining the moving direction of the intruder is also an important task in our exploration as the resulting pattern can help to direct further defense strategies, e.g., turning on the surveillance camera in one part of the floor or directing the law enforcement officers to follow the direction that the intruders go to. Figure 7 (c) presents the RSS readings for three transmitter-receiver pairs, T4-R4, T5-R5, and T5-R6 respectively when the experimenter walked from position A toward position E. In order to examine the changes of RSS clearly, we shifted the RSS readings by 20 dBm for T5-R5 and by 35 dBm for T5-R6. By combining the RSS readings from multiple sources, i.e., multiple transmitter-receiver pairs T4-R4, T5-R5 and T5-R6, we can determine the moving direction of the experimenter based on the moving pattern delay in time series. The moving direction can be further calculated as the positions of receivers are usually known and the locations of the transmitters can be localized easily using the traditional localization methods [1, 8].

4. Potential Applications

The research work in this subtask related to localization of sensors and robots has a high potential to be applied in location-aware military applications such as intruder detection, tracking, monitoring and geometric based protection. Further, the detection of co-moving objects can help to determine whether enemies are moving together or walking individuals so that to further infer the motives of enemy actions.

5. Project Assessment

Our work in this year has met the SOW objectives. The following is the related publications for our subtasks:

- Gayathri Chandrasekaran, Mesut Ergin, Marco Gruteser, Rich Martin, Jie Yang and Yingying Chen, "DECODE: Exploiting Shadow Fading to DEtect CO-Moving Wireless DEvices", IEEE Transactions on Mobile Computing (IEEE TMC), 2009.
- Jie Yang and Yingying Chen, "Indoor Localization Using Improved RSS-Based Lateration Methods", in Proceedings of IEEE Global Communications Conference 2009 Wireless Networking Symposium (Globecom 2009), Hawaii, USA, November, 2009.
- Gayathri Chandrasekaran, Mesut Ali Ergin, Jie Yang, Song Liu, Yingying Chen, Marco Gruteser and Richard Martin, "Empirical Evaluation of the Limits on Localization using Signal Strength", in Proceedings of the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2009), Rome, Italy, June 2009.

- Jie Yang, Yingying Chen, Victor Lawrence and Venkataraman Swaminathan, "Robust Wireless Localization to Attacks on Access Points", in Proceedings of the IEEE Sarnoff Symposium 2009, Princeton, NJ, April 2009.

6. Reference List

[1] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RFbased user location and tracking system,” in Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), March 2000, pp. 775–784.

[2] Y. Chen, J. Francisco, W. Trappe, and R. P. Martin, “A practical approach to landmark deployment for indoor localization,” in Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), September 2006.

[3] J. Yang and Y. Chen, “A theoretical analysis of wireless localization using RF-based fingerprint matching,” in Proceedings of the Fourth International Workshop on System Management Techniques, Processes, and Services (SMTPS), April 2008.

[4] Yingying Chen, Gayathri Chandrasekaran, Eiman Elnahrawy, John-Austen Francisco, Konstantinos Kleisouris, Xiaoyan Li, Richard P. Martin, Robert S. Moore, Begumhan Turgut, "GRAIL: A General Purpose Localization System," Sensor Review, special edition, Localisation Systems, Vol. 28, No. 2, pp.115-124, 2008.

[5] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” Signal Processing Magazine, IEEE, vol. 22, no. 4, pp. 54–69, July 2005.

[6] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system,” ACM Transactions on Information Systems, vol. 10, no. 1, pp. 91–102, Jan. 1992.

[7] N. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” in Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom), Aug 2000, pp. 32–43.

[8] D. Madigan, E. Elnahrawy, R. Martin, W. Ju, P. Krishnan, and A. Krishnakumar, Bayesian Indoor Positioning Systems,” in Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM), March 2005, pp. 324–331

[9] E. Elnahrawy, X. Li, and R. P. Martin, “The Limits of Localization Using Signal Strength: A Comparative Study,” in Proceedings of the First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), October 2004.

[10] K. G. Calkins, “E-Book, An Introduction to Statistics,” http://tinyurl.com/2jlrsu, 2008.

[11] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Addison-Wesley, 2005.

[12] M. C. Vuran, W. B. Akan, and I. F. Akyildiz, “Spatiotemporal correlation: theory and applications for wireless sensor networks,” Computer Networks, vol. 45, pp. 245–259, 2004.

Appendices

Appendix A: Statement of Work

A Heterogeneous Multi-Robot Multi-Sensor Platform for Intruder Detection

Objective:

In order to achieve autonomous deployment of mobile sensors such as service robots in tactical mobile sensor networks, it is critical that the service robots can obtain the position themselves and further to localize sensors, monitor their activities, and track the movements of sensors.

Further, mobile sensor/robot networks are more effective comparing to static sensor network, particularly for scenarios in dynamic environments. Mobile sensor/robot networks have the flexibility to reconfigure themselves according to dynamic changes of the environment they operated within. They can carry load and deliver load to desired positions, and charge themselves at a home station if necessary. However, how to program the mobile sensors/robots to achieve autonomous controllable mobility is an open problem that has received much attention recently. One of the objectives of our research is to develop effective decentralized control algorithms for mobile sensors/robots to formation, to coverage, and to reconfigure, while maintaining connectivity of the network considering sensors/robots have limited communication range.

After the deployment, a vast number of critical facilities must be protected against unauthorized intruders. A team of mobile robots working cooperatively can alleviate human resources and improve effectiveness from human fatigue and boredom. Since the robots can work autonomously, they are able to interpret sensor readings and recognize the intruders responsively, which can alert the human monitor of suspicious activity. In this way, the robot teams reduce manpower requirements while also increasing effectiveness. Based on the perception information, the robot can initiate a fast response to the situation by sending alert signals to a human operator, deploying a non-lethal weapon to capture the intruder, or autonomously tracking the movements of the intruder, etc.

In this proposed research, we will address the above challenges such as localization/tracking of service robots/sensor nodes, deployment and reconfiguration of mobile sensor/robot networks, and intruder detection. The ultimate goal is to develop an integrated multi-robot and multi-sensor test bed with the capability of localization, reconfiguration, and intruder detection.

Sub-Task2.1: Robot/sensor localization and tracking (Chen)

2.1.1 Localization of service robots/sensors: We will build a localization test bed with anchor-based approach. The service robots and sensors will be localized by a localization server. The server will report the position information periodically to the service robots.

2.1.2 Co-movement detection: With the ability of localization and mobility detection, we will investigate effective ways for object tracking. Especially we will study approaches to determine whether multiple robots/sensors are moving together.

2.1.3 Intruder detection using sensor nodes: Based on the variation in signal strength at sensor nodes caused by intruder movement, the localization server will determine abnormal changes and alert the service robots for possible intruders.

Final Report

W15QKN-05-D-0011, Task Order 43

(September 15, 2009)

Submitted by S. Tewksbury

Contract Number: W15QKN-05-D-0011, Task Order 43

Contract Name: Embedded Intelligent Sensor Network Systems

Task 2

Heterogeneous Multi-Robot Multi-Sensor Platform for Intruder Detection

Subtask 2

Robot deployment and effective decentralized control

Prof. Yi Guo
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030

Abstract

We developed decentralized patrolling algorithms for multi-robot systems. We proposed a new motion synchronization method and used it in designing the decentralized control laws. The goal is for each robot to move along a subsegment of equal length in equal time interval with potential impacts. The impact law depends only on the time information. Specifically, “the time interval between two consecutive impacts” is exchanged when the robots meet. We also show how to apply the synchronization algorithm to the planar patrolling problem. Simulation results show the feasibility and robustness of our algorithm. We started to implement the algorithm on E-puck mini robots.

1. Introduction

In order to achieve autonomous deployment of mobile sensors such as service robots in tactical mobile sensor networks, it is critical that the service robots can position themselves and further to localize sensors, monitor their activities, and track the movements of sensors. Further, mobile sensor/robot networks are more effective comparing to static sensor network, particularly for scenarios in dynamic environments. Mobile sensor/robot networks have the flexibility to reconfigure themselves according to dynamic changes of the environment they operated within. They can carry load and deliver load to desired positions, and charge themselves at a home station if necessary. However, how to program the mobile sensors/robots to achieve autonomous controllable mobility is an open problem that has received much attention recently. The objective of our research is to develop effective decentralized control algorithms for mobile sensors/robots to formation, to coverage, and to reconfigure, while maintaining connectivity of the network considering sensors/robots have limited communication range.

In this project, we developed a decentralized multi-robot patrolling algorithm. In particular, we first plan a complete coverage path, and then consider multi-robot system patrolling with potential impacts. We design impact laws (i.e.control laws when robots meet each other) to achieve motion synchronization by each robot moving along an equal-length subsegment in equal time-span on a line segment. The algorithm assumes simple information exchange, namely, the time span since the last impact, and assumes no knowledge of total number of robots, nor the total length of the line segment be known by the robots. While similar ideas appeared in literature, some distance measurement to critical points or priori knowledge such as the perimeter length or the total robot number is required. We relax such assumptions, and use only the information of robot interaction time and velocities in constructing the control laws. We also consider the scenario when multi-robot-impact (more than two robots) at the same point, which is ignored in previous work. Our algorithm is decentralized, and robots only communicate to their adjacent neighbors when they meet each other. It is robust to robot failures, in the sense that a removal or an addition of robots does not affect the patrolling goal and eventually every point of the patrolling path is visited with uniform frequency.

2. Approach Taken

We developed a decentralized control law to achieve synchronization in this project. The basic idea is that each robot in the system under motion moves in a constant velocity until

impact happens (i.e., when they meet). Then, we define different updating law when different type of impact happens. ``Constant velocity" means that the robot moves along a straight line without any changes of the magnitude and the direction of its velocity.

The flow chart of the algorithm is shown as in Figure 1.

The decentralized control laws are design to be

1. Face-face type updating law

$$g_1(q,t_p) \overset{\text{def}}{=} \begin{cases} v_i^+ = -sgn(v_i)(v_{ss} + a_1(\Delta t_i - \Delta t_{i+1})v_{ss}v_i) \\ v_{i+1}^+ = \\ \quad -sgn(v_{i+1})(v_{ss} + a_2(\Delta t_{i+1} - \Delta t_i)v_{ss}v_{i+1}) \\ \qquad\qquad\qquad\qquad q \in D_1 \end{cases}$$

2. Face-tail type updating law

$$g_2(q,t_p) \overset{\text{def}}{=} \begin{cases} v_i^+ = v_{i+1} \\ v_{i+1}^+ = -v_i \end{cases}, \quad q \in D_2$$

3. Hit-boundary type updating law

$$g_3(q,t_p) \overset{\text{def}}{=} v_i^+ = -sgn(v_i)v_{ss}, \quad q \in D_3$$

4. Multi-hit type updating law

$$g_4(q,t_p) \overset{\text{def}}{=} \begin{cases} v_i^+ = -v_i \\ v_{i+1}^+ = -v_{i+1} \\ \cdots \\ v_{i+k}^+ = -v_{i+k} \end{cases}, \quad q \in D_4$$

We apply the segment synchronization into a multi-robot area patrolling problem. Consider assigning an N homogeneous mobile robot system S to patrol a given 2D area, which has its patrolling interest uniformly distributed. We first partition the planar area into grids, and by finding a Hamiltonian path, we simplify the 2D patrolling problem into a 1D patrolling case. Patrolling in a 2D area is then converted to the problem of finding a Hamiltonian path. When a robot moves along the path, its sensor or effector covers the area eventually.

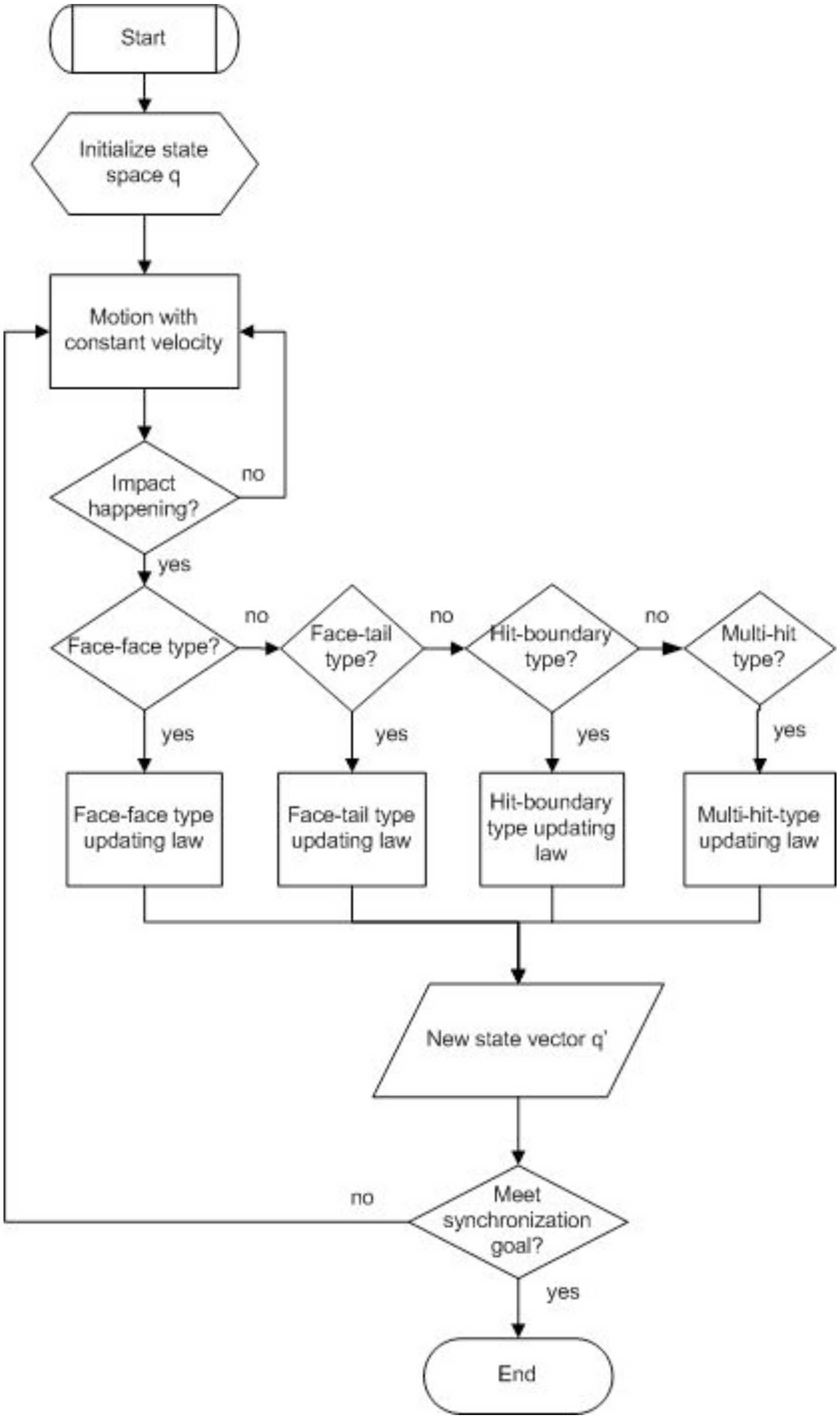

Start
Initialize state space q
Motion with constant velocity
Impact happening?
no
yes
Face-face type?
no
Face-tail type?
no
Hit-boundary type?
no
Multi-hit type?
yes
yes
yes
yes
Face-face type updating law
Face-tail type updating law
Hit-boundary type updating law
Multi-hit-type updating law
New state vector q'
no
Meet synchronization goal?
yes
End

Figure 1. Flowchart of the multi-robot synchronization algorithm.

3. Results

3.1. Matlab simulation results

We implemented the algorithm and tested in Matlab. We implemented a Spanning Tree Coverage (STC) method. We assume that a single robot is with a sensing range of D, then partition the area into cells that each cell has the size of $2D\times 2D$. Then, by building a spanning tree according to the cell size, a Hamiltonian cycle visits all cells of the domain by following the tree around. An illustration of STC method is shown in Figure 2, in which the dotted line is the spanning tree, the arrowed path is a Hamiltonian cycle around the spanning tree. Note that a Hamiltonian path can be generated from the Hamiltonian cycle by breaking the circle at any point.

The performance of the algorithm is shown in Figure 3, where an 6-robot system reaches synchronization on segment $[0,1]$. At time $t=0$, the position vector and velocity vector are [0.0960 \ 0.2843\ 0.3708\ 0.5275\ 0.5456\ 0.9811] and [-0.8706\ 0.0896\ 0.6728\ -0.7094\ -0.6570\ -0.8639] respectively. We choose the parameter in (9) as $a_1=0.92$, and $a_2=-0.84$. The system tends to reach synchronization by its trajectory uniformly distributing along the segment. Each subsegment is [0, 0.167], [0.167, 0.333], [0.333, 0.5], [0.5, 0.667], [0.667, 0.833], [0.833, 1], each robot moves along an equal length subsegment, back and forth at the same speed $v_{ss}=1$, which can be seen in the figure as the slope of each single short line is all the same at time $t=6$.

In Figure 4, we simulate the scenario that at time $t=$ 122.7s, a robot is suddenly taken out, which is illustrated as a vertical line from 0.5 to 0 at 122.7 sec. The other three robots will adapt to such dynamic change and reaches a new synchronization configuration by uniformly distributing along the segment, and the equal length subsegments are [0, 0.333], [0.333, 0.667], [0.667, 1].

In Figure 5, we demonstrate the case that 2 robots are added into the system at time point 202.4s, at $x_1=0.35$ and $x_2=0.6$ with the velocity $v_1=0.342$ and $v_2=-0.874$. It shows the system reaches a new synchronization configuration in about 15 seconds.

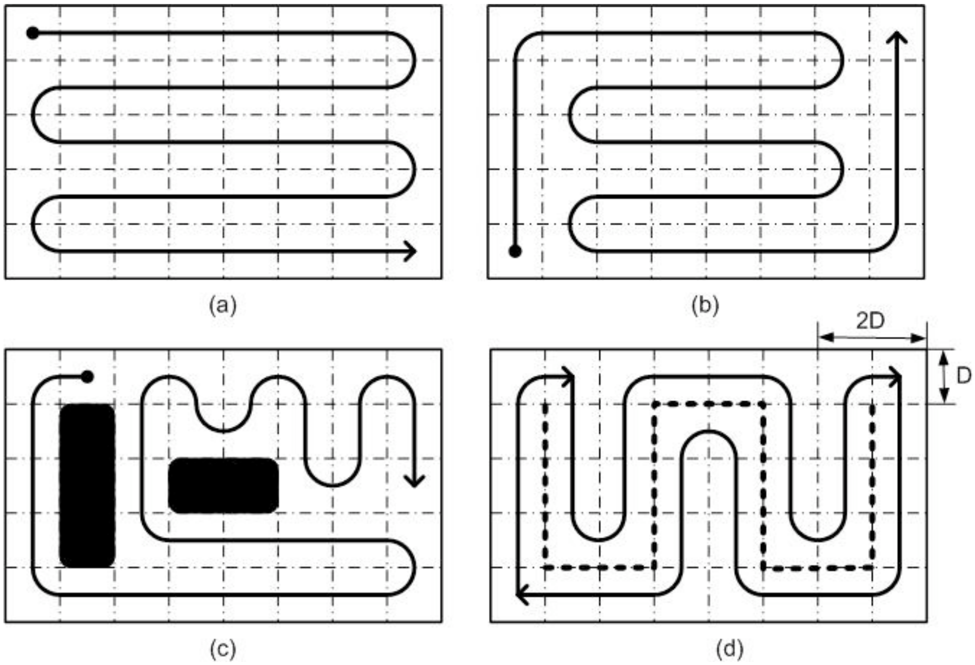


Figure 2. A series of illustrations of Hamiltonian path that covers the whole area. a) A Hamiltonian path illustration. b) Another Hamiltonian path illustration. c) A Hamiltonian path in the environment with obstacles. d) An illustration of Hamiltonian cycle generated by STC method.

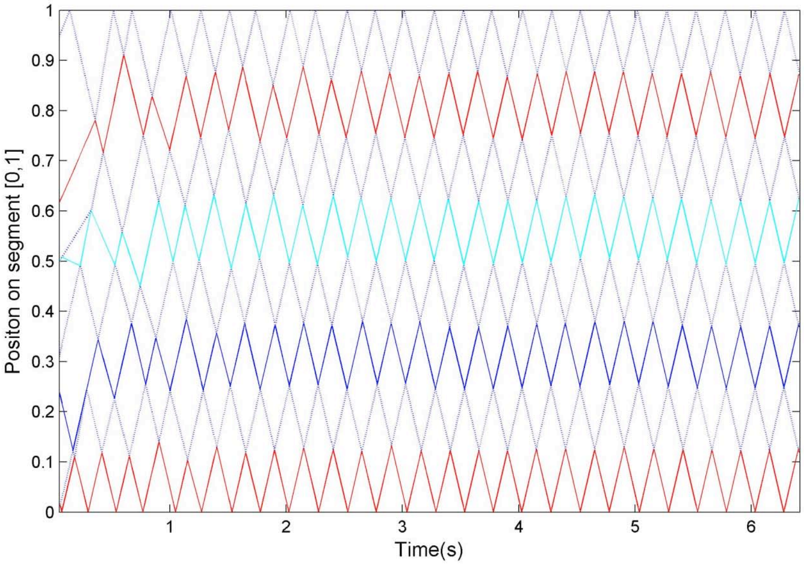


Figure 3. Simulation result of 8-robot system synchronization on the segment [0,1]

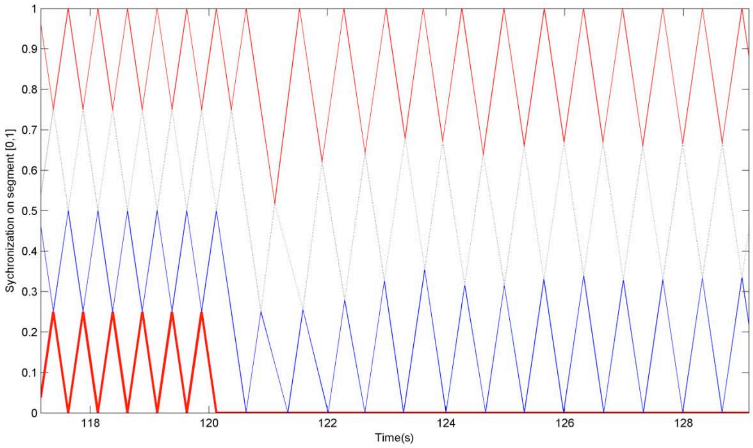


Figure 4. The system response when a robot is taken out at time t=122.7 sec

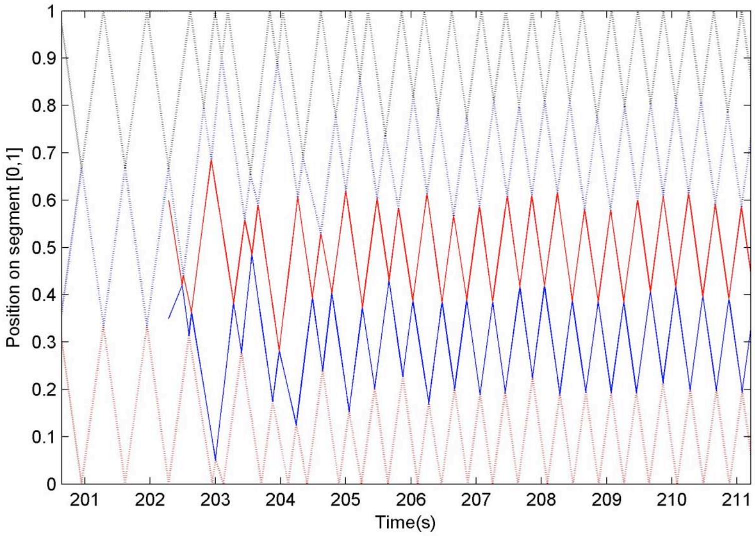


Figure 5. The system response when two other robots are added into the system at time t=202.4 sec

3.2. Webots simulation and E-puck robot experiments

We started to implement the algorithm in Webots. Webots is a software platform for fast prototyping and simulating of mobile robots, and it facilitates the transfer of the developed algorithm to the real mobile robots. We plan to use E-puck mini robots for real robot experiments. The planned patrolling scenarios are shown in Figures 6 and 7.

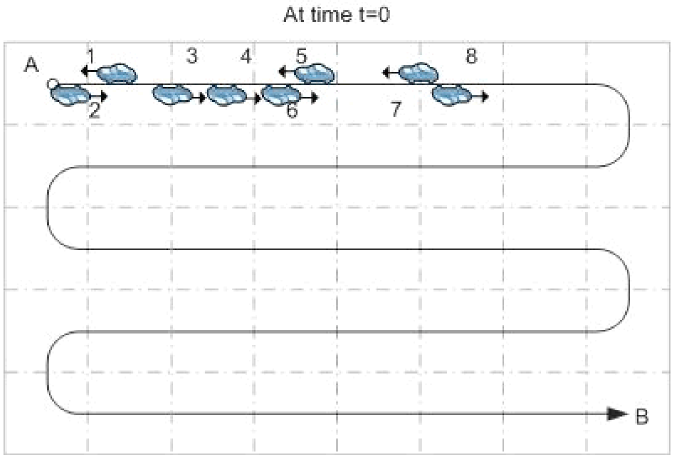
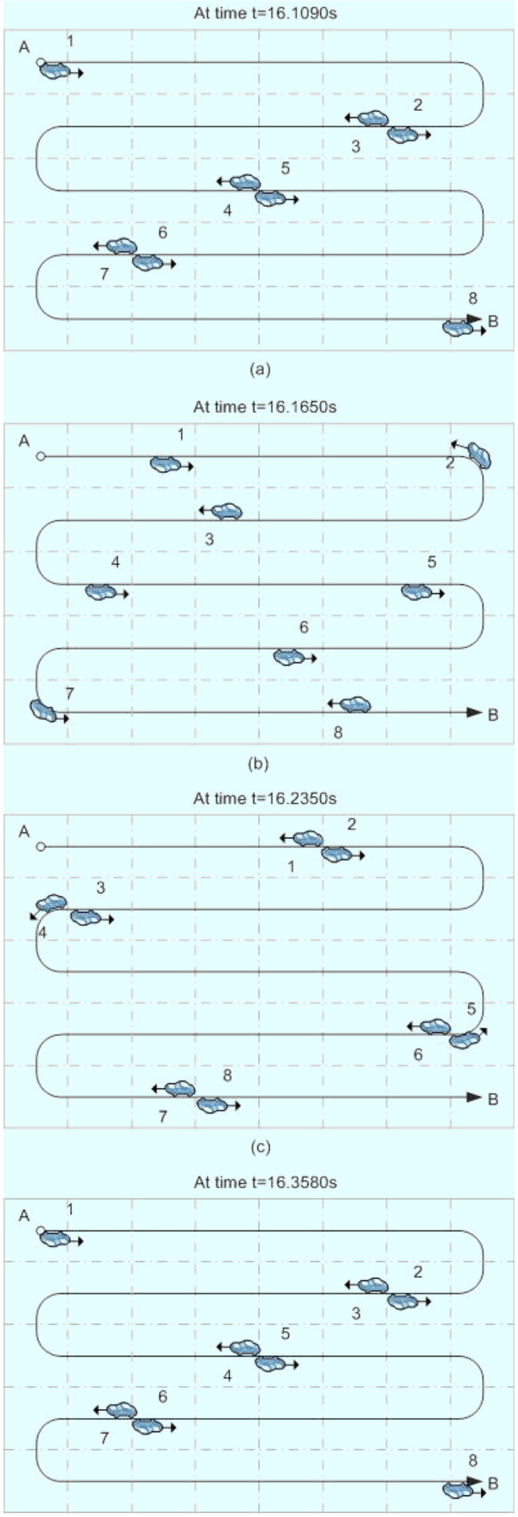


Figure 6. The initial configuration of mobile robot system



(a)

(b)

(c)

Figure 7. The patrolling path of a multi-robot system.

4. Potential Applications

In this project, we present a solution to multi-robot synchronization on a line segment with sporadic communication, which does not require any information on the localization of robot. Instead, the robot updates its velocity mainly based on the time span between two consecutive impacts of robot. We then apply the synchronization to a planar patrolling problem, based on the notion of a Hamiltonian path. Our solution guarantees that each point in the area is visited with a uniform frequency. Simulation results validate our algorithm, and show the efficiency and robustness of the method.

Potential applications of the results include autonomous deployment of multiple sensor and multiple robot networks for intruder detection.

5. Project Assessment

We have conduct research in developing decentralized deployment algorithms for multiple mobile robots in sensor network applications. The objectives relating to robot deployment proposed in the SOW have been met. We also in the process of implementing the algorithm on E-puck robot platform and tested it in our lab. A robot demonstration has been planned on Oct. 15 at Picatinny to run the algorithm on the real robots. Our future plan includes increasing the technology readiness lever of the project.

The project generates a few ideas for future work, which include dynamic coverage/formation control of multi-robot multi-sensor networks. Also, the relationship between coverage and connectivity to meet different application scenarios needs to be further investigated.

The project generates the following publication:

Hua Wang and Yi Guo, “Synchronization on a Segment Without Localization: Algorithm and Applications”, IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS), to appear, St. Louis, MO, Oct. 11-15, 2009.

6. Reference List

[1] M. B. Sevryuk, Estimate of the number of collisions of N elactic particles on a line, *Theoretical and Mathematical Physics*, 96(2003), pp 64-78.

[2] S. L. Glashow and L. Mittag, Three rods on a ring and the triangular billiard, *Journal of Statistical Physics*, Vol. 87, no. 3-4, 1997, pp. 937-941.

[3] B. Cooley and P. K. Newton, Iterated Impact Dynamics of N-Beads on a Ring, *SIAM Review*, Vol. 47 , Issue 2(2005), pp. 273-300.

[4] S. Susca and F. Bullo, Synchronization of beads on a ring, Decision and Control, 2007 46th IEEE Conference on , vol., no., pp.4845-4850, 12-14 Dec. 2007.

[5] D B. Kingston, R W. Beard and R Holt, Decentralized Perimeter Surveillance Using a Team of UAVs, *IEEE Transactions on Robotics*, vol. 24, No. 6, pp. 1394-1405, 2008.

[6] Y. Elmaliach, N.Agmon and G. A. Kaminka, Multi-robot area potrol under frequency constraints, *IEEE ICRA 2007*, Roma, Italy, 2007, pp. 385.

[7] K. Williams and J. Burdick, Multi-robot boundary coverage with plan revision, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, May 2006, 1716-1723.

[8] D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. Mclain, S.-M. Li, and R. Mehra, Cooperative forest fire surveillance using a team of small unmanned air vehicles, *International Journal of Systems Sciences*, vol. 37, no. 6, pp. 351-360, 2006.

[9] G. Chartrand, Introductory graph theory, *Courier Dover Publications*, 1985, ISBN 0486247759, 9780486247755, 294 pages.

[10] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31:77-98, 2001.

[11] N. Agmon, N. Hazon and G. A. Kaminka, Constructing spanning trees for efficient multi-robot coverage, *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, vol., no., pp. 1698-1703, May 15-19, 2006

[12] D. B. Kingston, Decentralized control of multiple UAVs for perimeter and target surveillance, *Doctor of Philosophy thesis*, Brigham Young University, December, 2007.

[13] Y. Zou and K. Chakrabarty, Sensor deployment and target localization based on virtual forces, *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol.2, no., pp. 1293-1303 vol.2, 30 March-3 April 2003.

[14] Y. Guo and M. Balakrishnan, Complete coverage control for non-holonomic mobile robots in dynamic environments, *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, May 2006, pp. 1704-1709.

Appendices

Appendix A: Statement of Work

2.2.A Heterogeneous Multi-Robot Multi-Sensor Platform for Intruder Detection

2.2.1. Scope

2.2.1.1.In this research, the contractor shall address the above challenges such as localization/tracking of service robots/sensor nodes, deployment and reconfiguration of mobile sensor/robot networks, and intruder detection. The ultimate goal is to develop an integrated multi-robot and multi-sensor test bed with the capability of localization, reconfiguration, and intruder detection.

2.2.1.2.The following are the proposed tasks:

2.2.1.2.1. Robot/sensor localization and tracking (Prof. Yingying Chen)

2.2.1.2.2. Robot deployment and effective decentralized control (Prof. Yi Guo)

2.2.1.2.3. Intruder detection (Prof. Yan Meng)

2.2.1.2.4. Integration of multi-robot and multi-sensor platform (The Team)

2.2.1.3.Robot deployment and effective decentralized control

2.2.1.3.1. The contractor shall develop effective deployment algorithms for the mobile robot team to cover a bounded area, and to reconfigure themselves when detecting intruders. The goal is to achieve maximum coverage of the robot team while maintaining connectivity of the robot network and avoiding collisions between team members.

2.2.1.3.2. The milestones include:

2.2.1.3.2.1.Decentralized deployment of mobile robots: effective decentralized deployment algorithms will be developed to ensure the robot network is always connected although the robots are in continuous motion.

2.2.1.3.2.2.Effective coverage control: A secondary objective including formation, coverage while maintaining connectivity will be investigated and algorithms will be developed to achieve the secondary objective. Constraints such as collision avoidance will be also considered.

2.2.1.3.2.3.Dynamic re-configurability: algorithms will be developed for the robot team to reconfigure themselves when detecting intruders.

2.2.1.4.Integration of multi-robot and multi-sensor platform. An integrated multi-sensor/multi-robot test bed will be developed in two phases. For phase 1 during the year of 2008-2009, a centralized localization server will be used to forward location information of sensor nodes and service robots to service robots, whereas the deployment, reconfiguration, and decision making of service robots are decentralized. For phase 2 during the year of 2009 – 2010, a totally decentralized integrated test bed will be implemented and demonstrated.

2.2.2. Deliverables

2.2.4.1. A comprehensive technical report of algorithms for four subtasks.

2.2.4.2. A localization test bed that can localize and track transmitters including service robots and sensor nodes in a laboratory environment.

2.2.4.3. A multi-robot test bed for autonomous deployment and effective decentralized control in a laboratory environment.

2.2.4.4. A multi-robot test bed for intruder detection in a laboratory environment.

2.2.4.5. Demonstration of an integrated test bed in a laboratory environment with a centralized localization server to forward location information to service robots, whereas the deployment, reconfiguration, and decision making of service robots are decentralized.

Appendix B: Proprietary Information

None.

Appendix C: Investigator-Sensitive Information

None.

Final Report

W15QKN-05-D-0011, Task Order 43

(September 15, 2009)

Submitted by S. Tewksbury

Contract Number: W15QKN-05-D-0011, Task Order 43

Contract Name: Embedded Intelligent Sensor Network Systems

Task 2

Heterogeneous Multi-Robot Multi-Sensor Platform for Intruder Detection

Subtask 3
Intruder detection

Prof. Yan Meng
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030

Part One: Dynamic Task Allocation among Robots

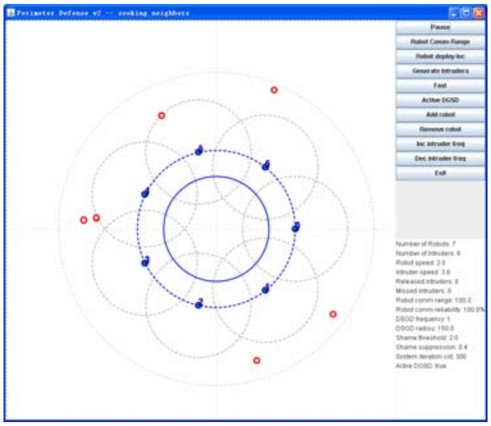
Abstract

In security defense tasks, multiple robots need work corporately to detect offensive intrusion to protect some sensitive areas. In this project, we propose a distributed algorithm for a multi-robot system with some static sensors. The system concept is that static sensors sense intrusions and act as a cueing sensor to an ensemble of robots. These robots in turn engage the potential intruder, performing surveillance and/or neutralization of the intrusion. To minimize the intruder missing rate and average response time, a STAGS (Shame-level Task Allocation and Gap-based Self-deployment) method is proposed, which is a decentralized method without a central control unit. To further improve the system adaptability under dynamic environments, a multi-objective optimization (MOO) method is proposed to adjust the system parameters of STAGS. Extensive simulation results demonstrate the effectiveness and robustness of the proposed algorithm in a dynamic intruder detection task.

1. Introduction

Security defense task is a complex problem, which aims to protect sensitive areas against offensive intrusion. Video surveillance system is one of the solutions for these tasks, which still require manned observation and can be quite costly for large areas. Another alternative solution is to use autonomous multi-robot systems (MRSs) for intruder detection to reduce the overall system cost without compromising security.

In this project, we will describe an autonomous system consisting of cooperative mobile robots with some static sensors for security defense tasks. The system would utilize many relatively cheap sensors that can be used as a cueing sensor for an ensemble of robots to detect and track the movements of intrusion of any kind through a predetermined area or boundary. Through the use of mobile robots, the intruders can be tracked, intercepted, or neutralized. While some robots are investigating the intruders, the remaining robots would self-deploy themselves to maximize coverage. Fig.1. illustrates our simulator for this problem.



Number of Robots: 7
Number of Intruders: 8
Robot speed: 2.0
Intruder speed: 3.0
Released intruders: 8
Missed intruders: 0
Robot comm range: 100.0
Robot comm reliability: 100.0%
DGSD frequency: 1
DGSD radius: 150.0
Shame threshold: 2.0
Shame suppression: 0.4
System iteration cnt: 306
Active DGSD: true

Fig. 1. A snapshot of the security defense problem simulator. The area to be protected is the blue solid circle. Seven robots are deployed on outer blue dotted circle (deployment circle). The communication range of each robot is represented by grey dotted circle. The red dots are intruders and blue dots are robots.

The objective of this system is to coordinate robots to minimize the *missing rate* and *average response* *time* to the intruders. Missing rate is the percentage of intruders which successfully invade the protected area without being investigated by robots over all the intruders. Response time is the time period from the time of an intruder is detected by sensors to the time it is investigated by robots. Intruders attack the protected area in a random manner which requires the robots to react in a real-time performance.

Extensive work has been proposed for multi-robot coordination for various applications, one paradigm is based on organization theory derived from human social behavior and psychology [1][5][10][16]. Another paradigm is bio-inspired algorithms [6][9][14]. Singh and Thayer [13] proposed a distributed multi-robot coordination method in a demining problem that mirrors the mechanism of the human immune system to modules of software architecture. Capability to learn unknown situations and react to the learned situations efficiently has been achieved. The model considers robots as B cells and mines as antigens, binding affinity between a robot and a mine is inverse proportional to the distance between them. When a robot finds a mine, it will stimulus nearby robots to come and help to diffuse the mine. Wu et al. [15] proposed an immune system based multi-robot exploration approach, where robots are setup as B cells and the locations of robots in unknown area are antigens. Based on robots' mutually stimulus and suppress, each robot picks up a destination in the unexplored area.

Very few works have been directly addressed for security defense problems [11][12]. Machado [12] proposed a distributed MRS approach for patrolling in a complex environment based on a market economy approach. In this project, we propose a STAGS (Shame-level Task Allocation and Gap-based Self-deployment) approach, which consists of a distributed shame-level based dynamic task allocation algorithm for intruder tracking and investigation, and a distributed gap-based self-deployment (DGSD) algorithm for self-deployment. Robots have to choose their own behaviors dynamically based on their current states and the environment. The parameters in STAGS approach need to be defined. To further improve the system robustness and adaptability to various environmental changes, a multi-objective optimization (MOO) method is applied to dynamically tune the parameters of the STAGS approach, where the two objectives are minimization of missing rate and average response time.

2. The Decentralized STAGS Approach

The STAGS approach consists of two parts: the first one is a shame-level based algorithm for dynamic task allocation, and the second one is the gap-based algorithm for self-deployment.

2.1 A Shame-Level based Dynamic Task Allocation Algorithm

Inspired by [8], a shame-level based algorithm is proposed to dynamically allocate robots to detected intruders. When sensors detect an intruder, the intruder's information is broadcasted to all the neighboring robots. Then each robot develops a shame level for this detected intruder, which is inversely proportional to their distance to the intruder. The shame level is incremented until it reaches a threshold that causes the robot to respond. Once a robot starts to respond to an intruder, the robot would broadcast its decision to its neighboring robots so that the neighboring robots would suppress their shame levels to this intruder. In essence, other robots no longer "feel" the shame of not responding to the intruder, so that they can investigate other intruders or self-deploy themselves.

The shame level of a robot r_i on intruder I_j can be defined as:

$$S(r_i, I_j) = \frac{\alpha \cdot v_i}{d(r_i, I_j)} \prod p(r_i, I_j) \tag{1}$$

where v_i is the robot speed. $d(r_i, I_j)$ is the traveling distance between r_i and I_j. α is a constant factor. $p(I_j, r_k)$ is the shame-level suppression for r_k on intruder I_j, which can be defined as:

$$p(I_j, r_k) = \begin{cases} 1 \text{ when } r_k \text{ is tracking } I_j \\ \beta \ (0<\beta<1) \text{ when } r_k \text{ is not tracking } I_j \end{cases} \quad (2)$$

where β is a constant representing the suppression level.

2.2. A Decentralized Gap-based Self-Deployment (DGSD) Algorithm

When some of the robots start tracking intruders, the rest of the robots should deploy themselves uniformly in the deployment circle to cover as much area as possible. A gap-based algorithm is proposed here for this self-deployment purpose. A gap is defined as the arc area generated by any two tracking robots and the center of the protected circle. The corresponding tracking robots are called gap builders, and robots within the gap is called gap members. The gap members should be deployed uniformly within each gap. Based on different situations of intruders and robots, a gap weight is assigned dynamically to each gap. Gap weight W_G for each gap can be defined as:

$$w_G = \frac{s_G + \beta * (n_{IG})}{n_{rG}} \quad (3)$$

where s_G is gap G's angle in degree. n_{IG} and n_{rG} are the number of intruders and robots within gap G, respectively. β is a constant that adjusts the importance of n_{IG}. A gap with a higher gap weight has a higher priority to cover. In other words, more deploying robots should join in the gaps with higher weights. Therefore, the objectives of gap-based method are: (1) deploy gap members uniformly within the gap; (2) switch gap members to a neighboring gap with a higher gap weight.

Each gap has a DGSD process which runs periodically. The DGSD process contains a round-trip to pass information to all the gap members within the gap. The round-trip starts from a gap builder R_B. R_B generates a information pack containing its local information. The pack is delivered to the other gap builder R_B' by passing through each gap member one by one locally. During the delivering process, information pack is updated with gap member's local information. So R_B' has a full view of current status of the gap, such as number of gap members, where gap starts and ends, and gap size, etc. Base on this information, R_B' is able to generate a proper deployment plan. The plan is delivered back to R_B through local passing agents one by one. It is worth to note that only local communication is needed for the robots for DGSD since the information is passed one by one instead of globally broadcasting.

Gap builders also hold status information of two neighboring gaps so that it can notify a gap member to switch to a neighboring gap if the neighboring gap has a much higher weight. In this manner, critical gaps will attract more robots.

Fig.2 shows an example for this DGSD process. In gap1, DGSD process is started by gap builder R_1. R_1 sends out the information pack to R_2, then R_2 sends the information to R_3. When R_4 (another gap builder) receives the information pack from R_3, it is notified that there are two gap members (robots) and three intruders in Gap1. Then R_4 updates the memory of w_{G1} and calculates proper deployment. Then, this deployment information is delivered back to R_1 through R_3 and R_2. As a result, R_1 updates the memory of w_{G1}, and R_2 R_3 deploy on stars. For other gaps, R_5 will switch to Gap1 from Gap2 because Gap1 has a higher weight. R_7 will stay in Gap3 to investigate intruder I_4.

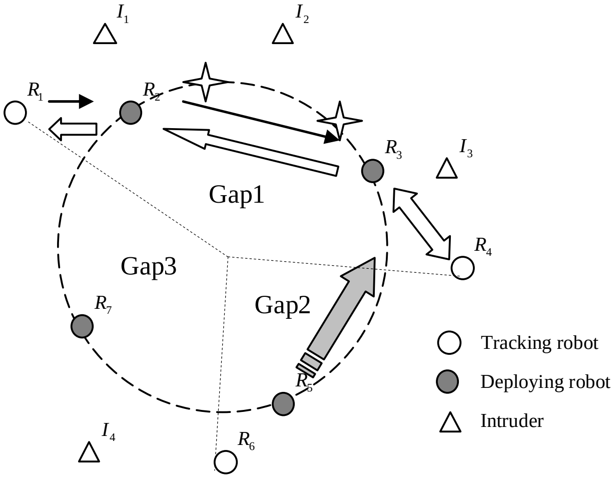


Fig. 2. One example using the DGSD algorithm.

If a deploying robot cannot communicate with its neighbors due to its limited communication range, for example, its neighbors move away, this deploying robot will move in the direction till a proper neighbor is found and the package delivery can be continued.

Fig.3 shows the block diagram of the STAGS algorithm. These two algorithms have mutual influence with each other. The shame-level based algorithm triggers robots to conduct intruder investigation. Meanwhile, the investigating robots would dynamically formulate gaps. With the new gaps, the robots use the DGSD algorithm to deploy themselves within the gaps to corporately working with the investigating robots, which would further affect the performance of future investigation.

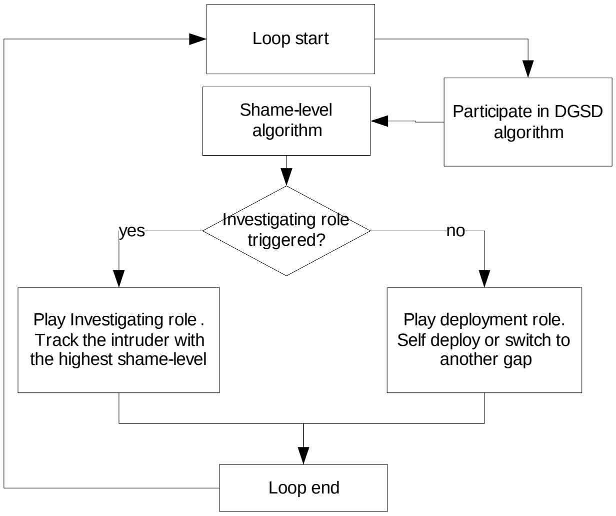

Loop start
Shame-level algorithm
Participate in DGSD algorithm
Investigating role triggered?
yes
no
Play Investigating role . Track the intruder with the highest shame-level
Play deployment role. Self deploy or switch to another gap
Loop end

Fig. 3. The block diagram of the STAGS algorithm.

2.3. Online Learning and Multi-Objective Optimization on Distributed STAGS Method

Since the parameters of STAGS method need to be defined, due to the dynamic intruder behaviors, it is hard to find optimal parameters for all different situations. Ideally, the solution should be self-adaptive, which requires the robot system to recognize and self-adjust for different situations. Meanwhile, the system should be capable to handle unknown situations with real-time performance. To achieve these two features, a multi-objective optimization (MOO) method is proposed to dynamically adjust the parameters of the proposed distributed control models.

Problem situation S can be defined as: *{intruders' arriving rate, intruder's speed, number of robots, robots' speed}*, where the intruders' arriving rate is the frequency of the arrivals of new intruders.

A linear approach of exactly matching is applied to estimate the situation difference. If we define a situation pattern as $S_i = \{s_1, s_2, ..., s_k\}$, where s_k are the parameters that describe situation *k*. The matching can be estimated by the following equation:

$$\text{Matching} = (\frac{(s_1 - s'_1)}{s_1} < d_1) \wedge (\frac{(s_2 - s'_2)}{s_2} < d_2) \wedge ... \wedge (\frac{(s_k - s'_k)}{s_k} < d_k) \qquad (4)$$

where d_k is the difference upper bound for s_k and is setup as 10% for all situation perimeters. The adjustment option A is defined as: A = {*shame-level threshold*, *shame-level suppression*, *deployment radius*}. Deployment radius is the radius of the deployment circle as shown in Fig.1.

To find an optimal set of parameters for STAGS method, an individual robot's performance not only depends on its own parameter set, but also on the parameter sets of other robots and intruders. Due to the dynamic environment, the parameter setting has to evolve with the current environment status. Therefore, an online learning method is proposed here. Connected with sensors, the necessary situation information and the adjusted parameters of the STAGS algorithm are sent to all robots. The learning process is evaluated based on two criteria: the intruder missing rate and the average response time to intruders. A good strategy should strike a balance between these two criteria.

This is a multi-objective optimization (MOO) problem, where the objective function is no longer a scalar value, but a vector. As a consequence, a number of Pareto-optimal solutions should be achieved instead of one single solution. NSGA-II [3] has been adopted for evolution, which is a popular and efficient evolutionary algorithm for solving multi-objective optimization problems. In our work, simulated binary crossover (SBX) [2] and polynomial mutation [4] have been employed to generate offspring. After the offspring population is generated, the elitist crowded non-dominated sorting is used for selecting parents for the next generation.

Different from single objective optimization algorithms, where often only one optimal solution is achieved, NSGA-II produces a set of Pareto-optimal solutions, i.e. in our case, the parameter sets that balance the intruder missing rate and the average response time, and then the parameter set with the lowest missing rate is selected as final adjustment option. We will analyze the solutions in discussing the simulation results using NSGA-II. The complexity of NSGA-II is $O(MN^2)$, where *M* is number of objectives and *N* is population size of each evolution.

For each situation, NSGA-II requires some time to generate a mature result. However, the learning situation may change before sufficient evolution is reached. To solve this problem, we further add a learning process protection and resume mechanism to the NSGA-II algorithm. Basically, when situation changes from S to S', before system starts learning S', the learning process to S is protected in memory, which can be resumed in the future when S happens again.

The pseudo code of the acquired immune layer is summarized as followings:

Step1. Detect environment changes periodically. Get current situation $S_{current}$.

Step2. Match $S_{current}$ with $S_{previous}$ which is the last detected environment situation. If the match is founded, go to step 4, otherwise, go to step 3.

Step3. If learning on $S_{previous}$ is not finished, protect $S_{previous}$'s learning process. Go to step 4.

Step4. If a { S_{ex} , A_{ex} } in memory matches $S_{current}$, use A_{ex} to adjust the STAGS parameters. Otherwise, go to step 5.

Step5. Start, continue or resume learning to $S_{current}$ using the NSGA-II method. When the learning process on $S_{current}$ is finished, store { $S_{current}$, $A_{current}$ } into memory.

3. Results

To evaluate the performance of the proposed model for intruder detection in a security defense task, a simulator is developed in Java, as shown in Fig. 1. The environment is a 800x800 square area. The protected area is defined as a circle with the diameter of 200. Sensors are deployed uniformly around the circle and can detect a circle with the diameter of 750. New intruders' initial locations are uniformly distributed on the boundary of the area that can be sensed by the sensors. It is assumed that new intruders appear following a Poisson distribution pattern as:

$$P(X=k)=\frac{\lambda^k e^{-\lambda}}{k!}(k=0,1,2,....) \qquad (5)$$

where $P(X=k)$ is the probability that *k* new intruders arrive in each simulator iteration (the simulator's basic time unit). The expectation of $P(X)$ is λ, so on average a new intruder appears on every $1/\lambda$ system iteration.

3.1. Simulation Results of STAGS with Fixed Predefined Parameters

To evaluate the performance of STAGS algorithm (S for shame-level based algorithm and D for gap-based self-deployment algorithm), two simple algorithms are defined here: numb tracking (NT) algorithm where robots always track the closest intruder, and numb deployment (ND) algorithm, where robots are initially distributed uniformly on the deployment circle and return to their initial locations when the investigation jobs are finished.

In this simulation, shame-level threshold=2.4, shame-level suppression=0.2, deployment-range=235, and $\beta=1$ for the gap weight. Multiple simulations are carried on for different algorithm combinations NT+ND, S+ND, T+D, S+D under different situations. The missing rate and the average response time are listed in Table I. The results illustrate that S+D algorithm outperforms others. Applying D algorithm brings little improvement when the team size is relatively small. This is because when the team size is smaller, robots have to track intruders most of the time so leave much less time to play self-deployment role. However, when the self-deployment time is longer enough, the advantage of applying two algorithms together becomes more obvious.

TABLE I
SIMULATION RESULTS

		NT+ND	NT+D	S+ND	S+D
RN=8, IR=8 RS=4, IS=3.0	MS	45.28%	46.96%	8.40%	6.72%
	RT	66.46	67.37	43.68	40.38
RN=8, IR=8 RS=4, IS=3.5	MS	57.03%	55.76%	17.54%	15.63%
	RT	62.11	62.14	44.29	42.45
RN=6, IR=8 RS=4, IS=3.0	MS	46.20%	45.14%	14.84%	14.34%
	RT	67.27	66.81	49.31	48.60
RN=6, IR=8 RS=4, IS=3.5	MS	58.18%	57.18%	23.50%	22.64%
	RT	62.91	62.72	48.88	48.75
RN=4, IR=8 RS=4, IS=3.0	MS	51.84%	50.40%	25.98%	25.81%
	RT	69.27	69.13	58.45	58.25
RN=4, IR=8 RS=4, IS=3.5	MS	58.94%	58.85%	36.00%	35.77%
	RT	65.29	63.49	55.37	55.34

3.2. Simulation Results of STAGS with MOO-based Online Learned Parameters

jMetel software package is implemented in the simulator to realize NSGA-II algorithm. jMetel is a Java-based framework aimed at facilitating the development and experiment for solving multi-objective optimization (MOO) problems [7]. In our experiment, we configure NSGA-II's evolution population size as 8.0 and the maximum evolutions as 8.0. Other parameters use default values in jMetel package: crossover probability is 0.9 and mutation probability is 1/(size of A) which is 0.33. Dynamic cases for the perimeter defense problem are listed in Table II. Each time step T equals to 50,000 simulator iteration.

To evaluate the system performance of STAGS method with the MOO-based online learned parameters, the simulation is conducted in two modes: predefined mode and MOO-learned mode. In order to provide a thorough comparison with the MOO-learned mode, we conducted multiple experiments of STAGS method with different predefined parameters. For each perimeters in A, three values are chosen in a reasonable range so that a total 27(3*3*3) experiments are conducted in predefined mode. The parameter ranges and values in predefined mode are defined as follows:

Shame-level threshold: 1.2, 2.4, 3.6 $\in$ *[0, 4]*

Shame-level suppression: 0.2, 0.6, 0.8 $\in$ *[0, 1]*

Deployment-range: 145, 235, 285 $\in$ *[100, 350].*

These perimeter ranges are also applied in the MOO-learned mode when NSGAII algorithm evolves to generate candidate parameter sets.

TABLE II
DYNAMIC CASES FOR THE SECURITY DEFENSE PROBLEM

	Environment change
T 0	Intruders' arriving rate = 8, Robot number = 8 Robot speed = 4 Intruder speed = 3
T 1	Decrease robot number by 1
T 2	Decrease robot speed by 0.5
T 3	Increase intruders' arriving rate by 3
T 4	Increase intruder speed by 0.5
T 5	Increase robot number by 1
T 6	Increase robot speed by 0.5
T 7	Decrease intruder speed by 3
T 8	Decrease intruders' arriving rate by 0.5
T 9~16	Repeat step 1~8
T 17~24	Repeat step 1~8
T 25	Decrease robot number by 2
T 26	No change
T 27	Increase robot number by 2
T 28	Decrease robot number by 1

Fig.5. and Fig.6. show the simulation results of the intruder missing rate and average response time to the intruders for one case of the MOO-learned mode and 27 cases of the predefined mode. The results indicate that the performance of the MOO-learned mode is much better than all the experiments in the predefined mode on both criteria. Some experiments in the predefined mode that perform closely to the MOO-learned mode are particularly studied. The main reasons for this are: (1) as a genetic approach, NSGA-II may end up with some local minimum sometimes; (2) some algorithm parameter sets can handle the test cases very well for some specific situations. For example, sometimes we found parameter sets with lower shame-suppression may perform better than others. However, this does not necessarily mean that those parameter sets are able to handle all possible situations efficiently. On the other hand, the MOO-learned mode can automatically self-adjust those parameter sets through self-learning.

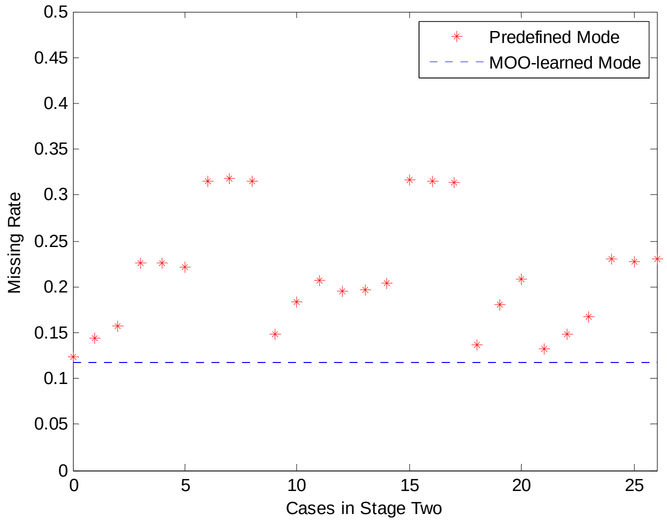


Fig.5. The intruder missing rates of one case using MOO-learned mode and 27 cases using predefined mode.

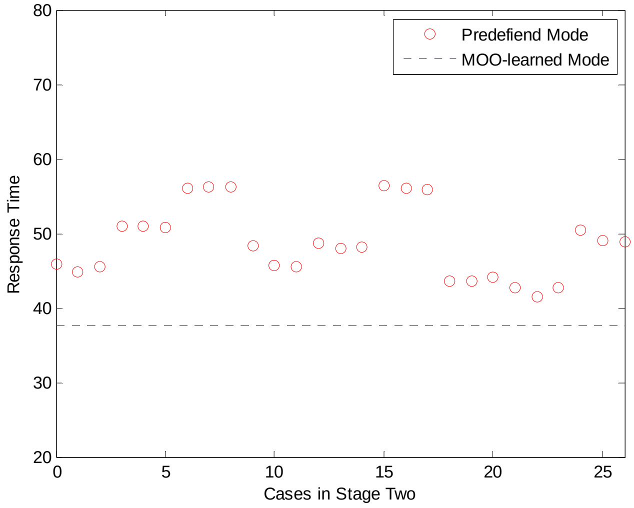


Fig. 6. The average response time of one case using MOO-learned mode and 27 cases using predefined mode.

To support this statement, further simulations are conducted on a randomly changing environment. Situations are chosen randomly from 27 possible combinations of situation parameters, which are listed as followings:

Robot number: 6,8,10

Intruder coming rate: 6, 8, 10

Robot speed/Intruder speed: 3/2.5, 3.5/3, 4./3.5.

Fig.7. and Fig.8. show the simulation results of the MOO-learned mode and the best case using predefined mode for missing rate and average response time, respectively. The best case in predefined mode is selected from previous simulation, with the parameters of {*Shame-level threshold: 2.4, Shame-level suppression: 0.2*, *Deployment-range: 235}*. Obviously, the MOO-learned mode still outperforms the best case using predefined mode. In addition, more experiences can be learned by the MOO-learned mode over time, which means that the advantage of MOO-learned mode would become more significant over time compared with the other mode.

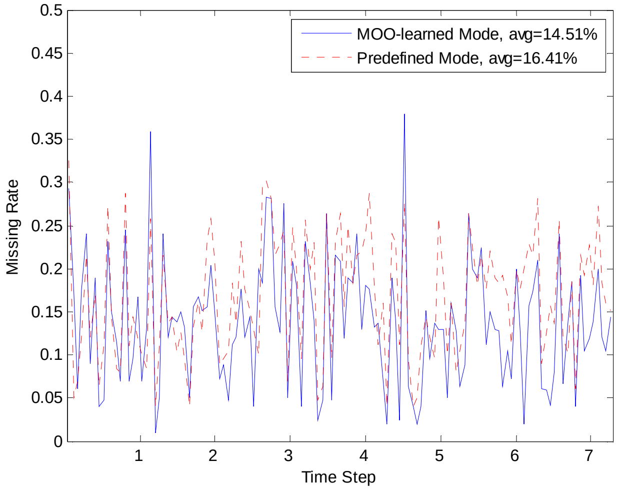


Fig.7. The comparison of the intruder missing rates under a random changing environment using both modes.

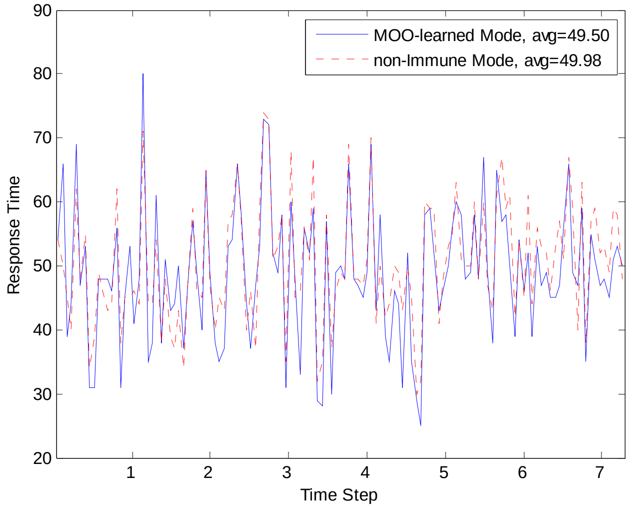


Fig.8. The comparison of the average response time under a random changing environment using both modes.

4. Potential Applications

In this project, we propose a STAGS algorithm for intruder detection in complex security defense tasks. A shame-based approach is developed for dynamic task allocation among robots to track the detected intruders, and a gap-based method is developed for the self-deployment of remaining robots. This STAGS algorithm is truly distributed, where only local communication among robots are needed and robots make their movement decisions only based on their local contextual information. To further improve the system robustness and adaptation, a MOO-based online learning method is developed to dynamically adjust the parameters of the STAGS method. The potential applications of the STAGS algorithm include situation awareness, security defense task, perimeter defense tasks, and security surveillance systems.

5. Project Assessment

This project has basically met the SOW objective, and the real world demonstration using the robotic platform will be conducted in an indoor environment to show the efficiency and robustness of the proposed approach for security defense tasks.

The following papers have been published or submitted based on this project.

1. Y. Zhang and Y. Meng, Dynamic Multi-Robot Task Allocation for Intruder Detection, 2009 *IEEE International Conference on Information and Automation (ICIA 09).* June 22-25, 2009, Zhuhai/Macau, China. (Finalist of best paper awards)
2. Y. Zhang and Y. Meng, A Decentralized Multi-Robot System for Perimeter Defense, *2010 IEEE International Conference on Robotics and Automation*. (submitted)
3. Y. Zhang and Y. Meng, STAGS: A Distributed Multi-Robot Cooperation Approach for Complex Security Defense Tasks, *Journal of Intelligence and Robotic Systems*, 2009.(submitted)

6. Reference List

[1] S.C.Botelho and R. Alami, “M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement,” *In Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 1234–1239.

[2] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” in *Complex Syst.*, Apr. 1995, vol. 9, pp. 115–148.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” in *Parallel Problem Solving from Nature (PPSN VI), M. Schoenauer et al., Eds. Berlin, Germany: Springer*, 2000, pp. 849–858.

[4] K. Deb and M. Goyal, “A combined genetic adaptive search (GeneAs) for engineering design,” *Comput. Sci. and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[5] B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, *Tech. Rep. CMU-RI-TR-05-13*, April 2005.

[6] M. Dorigo, E. Bonabeaub, G. Theraulaz, “Ant algorithms and stigmergy”, *Future Generation Computer Systems*, 16, pp. 851–871. 2000.

[7] J.J. Durillo, A.J. Nebro, F. Luna, B. Dorronsoro and E. Alba, “JMetel - A Framework for Multi-Objective Optimization,” http://jmetal.sourceforge.net/

[8] A. Gage, R. Murphy, K. Valavanis, M. Long, “Affective Task Allocation for Distributed Multi Robot Teams.” *CRASAR-TR2004-26*.

[9] Y. Gao, W. Wei, “Multi-Robot Autonomous Cooperation Integrated with Immune Based Dynamic Task Allocation”, *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06).*

[10] B. P. Gerkey and M. J. Matari´c, “Sold! Auction methods for multirobotCoordination”, *IEEE Transactions on Robotics and Autonomous Systems*, 18(5):758–768, October 2002.

[11] D.B. Kingston, R. Holt, R.W. Beard, T. McLain, and D. Casbeer, “Decentralized perimeter surveillance using a team of UAVs,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, August 2005, AIAA 2005-5831.

[12] A. Machado, G. Ramalho, J. Zucker and A. Drogoul, “Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures,” *Multi-Agent Based Simulation (MABS’2002)*, Bologna, 2002.

[13] S. Singh and S. Thayer, “Immunology Directed Methods for Distributed Robotics: A Novel Immunity-Based Architecture for Robust Control & Coordination” *SPIE: Mobile Robots XVI*, v. 4573.2001

[14] G. Wang, W. Gong, R. Kastner, “System Level Partitioning for Programmable Platforms Using the Ant Colony Optimization”, *13th International Workshop on Logic and Synthesis(IWLS’04)*, June 2004.

[15] H. Wu, G. Tian and B. Huang, “Multi-robot Collaboration Exploration Based on Immune Network Model,” in *Proc. 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1207-1212.

[16] R. Zlot and A. Stentz, “Market-based multi-robot coordination for complex tasks”, *International Journal of Robotics Research*, 25(1), January 2006, pp73-101.

Part Two: Intruder Recognition and Tracking

Abstract

In this project, a multi-layer local constrained hierarchical network (LCHN) is proposed to represent the features for visual object appearance. The connections of each node in this network are constrained by the local neighborhood of the node, which reflect the topology and dependencies of different parts of the object. Compared with a fully-connected network, the number of connections in LCHN is reduced while keeping spatial relationships of nodes. By applying a learning algorithm of minimizing contrastive divergence, this LCHN based model is able to learn complex feature structures from unlabelled data. More specifically, this model can provide hierarchical feature structures of the object of interest. The lower layer expresses more detailed appearance features while the higher layer represents more compact and abstract features. The experimental results demonstrate the efficiency of the learning capability of the proposed model and the feature hierarchy from the model for reconstruction.

1. Introduction

Learning and recognition of visual objects is a key problem in robot vision for various robot applications, such as robot navigation, search and rescue, service robots, etc.. Typically two steps are involved for object recognition: the computation of a set of target features and the combination of these features. Template-based approaches exhibit excellent performance in the detection of a single object, including faces [1], cars and people [2]. However, for more generic object recognition with many objects, large inter-class and intra-class variations pose big challenges for efficient learning and recognition. Therefore, many methods have been proposed to study more robust feature structures for object representation and recognition. The part-based models like constellation model [3] represent the geometric relationship among different parts of the interested object. But the correspondence hypotheses number is usually large, which leads to expensive computational cost. The models based on “bag of words” [4] [5] focus on learning the probability distribution of object parts as well as their dependency without considering the spatial connections.

On the other hand, many statistic learning methods have been applied to capture the hidden structures of object features for classifications. A linear SVM-based algorithm is proposed in [6] to automatically learn the discriminative components of face images. Principal components analysis (PCA) is applied in [7] to extract important features for online object learning and recognition. K-nearest neighbor (KNN) method is proposed in [8] to classify objects based on their distances in the feature space. However, this method would become intractable when the dimension of features is large. Latent Dirichlet allocation (LDA) based method [9] tries to find the latent variables behind the data, but usually the model needs to be crafted carefully.

Hierarchical approaches to represent objects have become increasingly popular recently, which are inspired by the hierarchical nature of human visual cortex. According to Hubel and Wiesel's theory [10], the cortex of human beings actually has hierarchical characteristics. The cortex consists of multiple levels with varied complexities. The bottom is simple cells that capture visual information from the environments directly. The processed signals are passed to the upper level consisting of complex cells, and then continue to hyper-complex cells. Each level deals with different complexities that represent different levels of the understanding about the environments. It is worthwhile to mention that the layer-wise connections are local, which means that a cell of the upper layer can only receive information from one group of cells of the lower layer instead of all groups from the lower layer.

The concept of feature hierarchy using multi-layer networks has been proposed in some work. In [11], a two-level feature set is obtained by combining position- and scale-tolerant edge-detectors over neighboring positions and multiple orientations, and a 'Standard Model' is proposed to stack multiple level features. However, most multi-layer networks usually need supervised learning with huge labeled data, which is not always feasible for objects with many categories. Similarly in [12], two-layer feature architectures are constructed, and then features are clustered in a high-dimensional space for object classification. In [13], the compositionality of visual object is represented by probability distributions, and the composition relation, shape features, and class categorizations are fused together in a Bayesian network for object classification. Recently, several works demonstrate the advantages of training multi-layer networks using unsupervised learning methods. An energy-constrained learning algorithm [14] is used for multi-level encoder-decoder networks, and invariant feature hierarchies are learned for object recognition. The contrastive-divergence-based learning is proposed in [15] to train the deep belief networks and Restricted Boltzmann Machine (RBM).

Inspired by the hierarchical architecture of visual cortex, in this paper, we propose a local constrained hierarchical network (LCHN) based model to learn the feature structures of objects. The LCHN consists of multiple layers and each layer represents different levels of features. The bottom layer works as the perception layer, which captures the most basic visual features. Then through the spatial-constrained connections between the bottom layer and its upper layer, the upper layer changes its own values according to the variations of the bottom layer. By passing this procedure upward, all layers adopt the variations of objects layer by layer. By applying the layer-wise learning algorithm proposed by Hinton [15], the hidden structures behind the visual features can be captured and expressed among the network. The proposed LCHN has the following advantages: 1) The spatial relationships and dependencies between object parts can be embedded into the network structures; 2) The hierarchy of the network provides different levels of descriptions on object features, which is usually very difficult for most algorithms of feature organization; 3) The unsupervised learning algorithm is applied for the network with the unlabelled data of different classes.

2. Approach Taken

2.1. The Local Constrained Hierarchical Network (LCHN)

Inspired by the hierarchical structure of visual cortex, a local constrained hierarchical network (LCHN) is constructed, as shown in Fig.1.

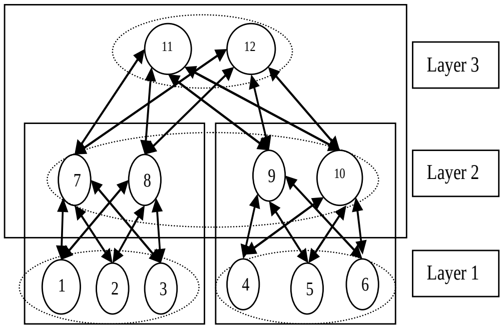


Fig. 1. An example of a 3-layer LCHN. Local constraints are circulated by ellipses with dashed lines. Three pairs of are contained in this model using rectangles and each pair is trained as a Restricted Boltzmann Machine.

This example network contains 3 layers with 6, 4 and 2 nodes for layer 1, layer 2, and layer 3 (from bottom to top), respectively. In layer 1, six nodes belong to two different local neighborhoods (the ellipses with dashed lines) and each local neighborhood has fully-connected mapping with its two upper nodes. In the middle layer, all nodes share the same local neighborhood and have connections with nodes of the top layer. The local neighborhood constraints reflect the relations among nodes. For example, if the neighborhood is decided by spatial relations of object patches, then patches 1-3 in layer 1 are constrained by the latent nodes 7 and 8 of layer 2. Patches 4-6 are decided by latent nodes 9 and 10. There is no connection between two groups in layer 1. However, all latent nodes of layer 2 belong to a single neighborhood, which reflects the spatial relations across the neighborhoods of the lower layer.

Based on the above example, we proposed the following rules of constructing the LCHN:

1) The LCHN has multiple layers. The number of layers depends on the problem. More layers mean more computational cost. For most object recognition problems, a network with 3 or 4 layers suffices.

2) A layer only has connections with its adjacent upper and lower layers. The bottom layer is only connected with its next upper layer and the top layer is only connected with its next lower layer. The layer-wise connections are constrained by local neighborhoods of each pair of layers.

3) The local neighborhoods can be determined by spatial distance, different features, or other factors. For visual objects, spatial distance is a good measurement for dividing local neighborhoods. The nodes of the same neighborhood have the same latent nodes of the upper layer. Different neighborhoods have no overlaps in the upper layer. In such a way, the number of connections is largely reduced compared with a fully-connected network. And neighborhoods of different layers reflect different scales of spatial relationships. The top layer with a single neighborhood represents the global features.

4) The network is undirected with symmetric connections between layers. From the cortex system point of view, the bottom-up propagation is of learning or perception, which captures different levels of knowledge from the observation. While the top-down propagation is similar to inferences or imaginations, which estimates the observation from experiences.

5) Once we have the network, next question is how to train the network to learn the patterns from the real data. Basically, the state of a network depends on the values of nodes and the connection weights of the network. Training a network means adjusting the values of nodes as well as the connection weights of the network through a learning algorithm using real data. Generally it is difficult and computational extensive to train a multi-layer network as a whole. However the LCHN can be decomposed into a number pairs of layers according to their neighborhoods and each pair can be trained independently. As shown in Fig.1, three pairs are marked by rectangles. Consequently the whole learning process can be divided into several small independent learning tasks.

For each pair, the lower layer can be called visible layer represented by V and the higher layer can be called the hidden layer represented by H. The state of a pair of layers can be defined as:

$$S(V,H)=\sum_{g=1}^{N}\sum_{i\in g}\sum_{j=g} s_{g,i}^{V} s_{g,j}^{H} w_{g,ij}^{VH} , \tag{1}$$

where $S(V,H)$ is the joint state of the visible layer *V* and hidden layer *H*. $s_{g,i}^{V}$ is the value of node *i* of neighborhood *g* in the visible layer. $s_{g,j}^{H}$ is the weight of node *j* of the hidden layer that is connected with neighborhood g. $w_{g,ij}^{VH}$ is the connection weight between these two nodes. The overall state is the sum of all connection pairs over all neighborhoods. Equation (1) can be further written as:

$$S(V,H)=\sum_{g=1}^{N}\sum_{i,j\in g} s_{i}^{V} s_{j}^{H} w_{ij}^{VH} \tag{2}$$

where $s_{i}^{V} s_{j}^{H}$ represents the connection pair that belongs to the same neighborhood *g*. If both $s_{i}^{V} s_{j}^{H}$ are binary stochastic units, this pair turns into a Restricted Boltzmann Machine (RBM), which can be trained by an unsupervised learning algorithm of minimizing contrastive divergence.

2.2. Restricted Boltzmann Machine

In the proposed LCHN, the layer pairs of each neighborhood can be treated as an RBM if all nodes have stochastic binary values. There are two layers for an RBM: a visible layer *V* and a hidden layer *H*. The state of an RBM can be defined by:

$$E(V,H)=-\sum_{i\in V} b_i v_i - \sum_{j\in H} b_j h_j - \sum_{i,j} v_i h_j w_{ij} , \tag{3}$$

where v_i and h_j represent the binary states of visible node *i* and hidden node *j*, respectively. w_{ij} is the connection weight between node *i* and node *j*. b_i and b_j are bias parameters. Given a data vector *V*, the hidden node *j* will turn into 1 with the probability of

$$p(h_j = 1 \mid V) = 1/(1+\exp[-(b_j + \sum_i v_i w_{ij})]) \tag{4}$$

Now the states of both visible and hidden nodes come from real data. Based on the values of hidden nodes given by (4), the visible data states can be recalculated or estimated as:

$$p(v_i = 1 \mid H) = 1/(1+\exp[-(b_i + \sum_j h_j w_{ij})]) \tag{5}$$

These values in equation (5) for visible nodes are reconstructed by the network, called reconstruction data. Using reconstruction data, the reconstructed hidden nodes can be calculated by applying equation (4) again. Now there are two sets of network states: the real data and the reconstruction data. Connection weights can be updated as:

$$\Delta w_{ij} = \varepsilon(\left\langle v_i h_j \right\rangle_{data} - \left\langle v_i h_j \right\rangle_{recon})$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \tag{6}$$

Where Δw_{ij} represents the change of the connection weight w_{ij}. ε is the learning rate. $\left\langle v_i h_j \right\rangle_{data}$ and $\left\langle v_i h_j \right\rangle_{recon}$ are the configuration products of visible and hidden nodes for real data and reconstruction data, respectively. Applying the learning rule of equation (6) to update the connection weights, the network will converge to the real data distribution. The similar rules can be applied for biases updates. This greedy learning algorithm is proposed by Hinton [16] and has been proved being efficient even though it is not strictly following the gradient of the log probability of the real data.

2.3. Training LCHN Using RBMs

By modeling the connection pairs of the same neighborhood using RBMs, the proposed LCHN model turns into the stacks and combinations of RBMs. The LCHN model of Fig. 1 can be decomposed into 3 RBMs. Therefore, by training the RBMs one by one, the whole network can be trained. Since there is no overlap between different neighborhoods of the same layer, the RBMs of the same layer can be trained simultaneously. Once the lower layer is finished, the training procedure can move up to the upper layer. This procedure continues until the top layer is trained. When the training moves up, the previous hidden layer turns into the visible layer. Also new neighborhoods are constructed based on the new neighborhoods. However, the learning procedure for RBMs is still the same. This procedure continues until the whole network is trained.

2.4. Extend LCHN with Inter-Node Dependencies (LCHN-ID)

The proposed LCHN model can be extended by adding inter-node dependencies, called LCHN-ID. The nodes that belong to the same neighborhood usually have inter-node connections. For example, as shown in Fig. 2, suppose node 1-3 of layer 1 represent object patches, the connections between these nodes represent their inter dependencies. More specifically, the dependencies between the nodes represent how likely these patches will be observed together in the object.

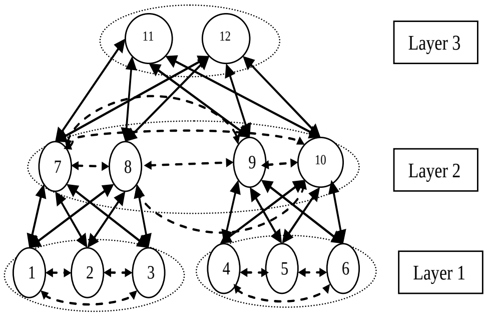


Fig. 2 The 3-layer hierarchical network with neighborhood constrains and inter-nodes dependencies (LCHN-ID).

After adding the inter-nodes dependencies, the energy of the corresponding RBM can be defined as:

$$E(V,H) = -\sum_{i\in V} b_i v_i - \sum_{j\in H} b_j h_j - \sum_{i,j} v_i h_j w_{ij} - \sum_{i\neq k} v_i v_k l_{ik} \qquad (7)$$

All variables have the same definitions with Equation (3) and l_{ik} represents the connections between the visible nodes. However, there is no connection between the hidden nodes. Otherwise, the learning strategy of RBMs is not valid. But if the hidden nodes turn into visible nodes in the RBM of the upper layers, they can be connected with each other. For example, as shown in Fig. 2, node 7 and 8 of layer 2 are independent when processing the RBM consisting of layer 1 and 2. And they become connected when the RBM of layer 2 and 3 is calculated since they become local neighbors. Similarly, the inter-node connections can be updated by equation (8).

$$\Delta l_{ik} = \varepsilon(\langle v_i v_k \rangle_{data} - \langle v_i v_k \rangle_{recon})$$
$$l_{ik} = l_{ik} + \Delta l_{ik} \tag{8}$$

where ε is the learning rate. $\langle v_i v_k \rangle_{data}$ and $\langle v_i v_k \rangle_{recon}$ are the configuration products of visible nodes for real data and reconstruction data, respectively.

3. Results

The proposed LCHN model is evaluated on the MNIST [17] database of handwriting digits including 60,000 training images and 10,000 test images. A four-layer network is constructed to learn the images. The bottom layer contains 28-by-28 784 nodes, which is the size of images, where each node corresponds to a pixel. All nodes are divided into 7-by-7 49 cells with each cell containing 4-by-4 16 neighboring pixels. The second layer keeps the same number of cells while each cell contains 3-by-3 9 nodes, which leads to the size of 441 nodes. Similarly, the third layer has 196 nodes with each cell having 2-by-2 nodes and the top layer has 49 cells with only 1 node inside a cell. So each layer has 49 cells with different number of nodes. The nodes population decreases from lower layers to higher ones since it is believed that the higher layers represent more abstract features with fewer nodes.

Then the neighborhoods for each layer are generated. For the bottom layer, each cell is a neighborhood which leads to 49 neighborhoods. If every 4 close neighborhoods merge into the same neighborhood of the upper layer, then the second layer has 16 neighborhoods and each neighborhood has either 3 or 4 cells. Similarly, the third layer has 4 neighborhoods and the top layer only has a single neighborhood.

Once the neighborhoods of different layers are created, the cells of corresponding neighborhoods can be connected. The initial weights of connections can be random numbers. Firstly, the basic LCHN is trained by using the data of MNIST database. Only 10% size of the database is used, i.e. 6000 training data and 1000 testing data. The training data is divided into 60 trunks evenly. Each trunk with 100 data is fed into the network as a whole for one training procedure. After training, the network takes the testing data as inputs and generates reconstructions. Then, the back propagation (BP) is applied to tune the network to get more precise reconstructions.

Fig. 3 shows the reconstructions generated by LCHN before and after 50 times BP. The top image is the true data. The left side is the outputs of all layers from top to bottom before the BP tuning. The right side is the outputs after BP. Before the BP, the LCHN can roughly reconstruct the shape of input digits, although the quality is not good. But after BP, much better reconstructions can be achieved.

The same procedure is applied to LCHN-ID with the same environments. Fig. 4 shows that LCHN-ID can provide better reconstruction results. The left-bottom image is the reconstruction of LCHN-ID before BP, which is much clear than the same image provided by LCHN in Fig. 3.

However, after fine tunings by a number of BPs, both models generate very similar results, as shown in the right-bottom images in Fig. 3 and Fig. 4. One reason to explain it is that after fine tunings, both models have been very close to the real pattern of the test data.

Then a fully-connected RBM network is tested on the same data. This network has 5 layers with 784, 1000, 500, 250 and 30 nodes. All nodes of each layer are fully connected with its upper and lower layers. Each layer pair consists of an RBM as well. Fig. 5 shows the reconstructions of this network. After BP tuning, the similar results are obtained.

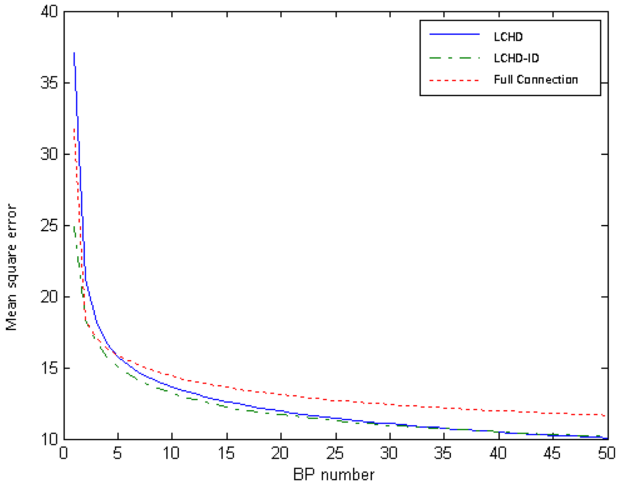


Fig.6. The mean square errors of reconstructions on testing data.

Fig. 6 shows the mean square errors of the reconstructions using three different models. It can be seen that the LCHN-ID provides the best starting point to tune the network. However, LCHN has the fewest number of nodes and connections, which leads to the fastest computation.

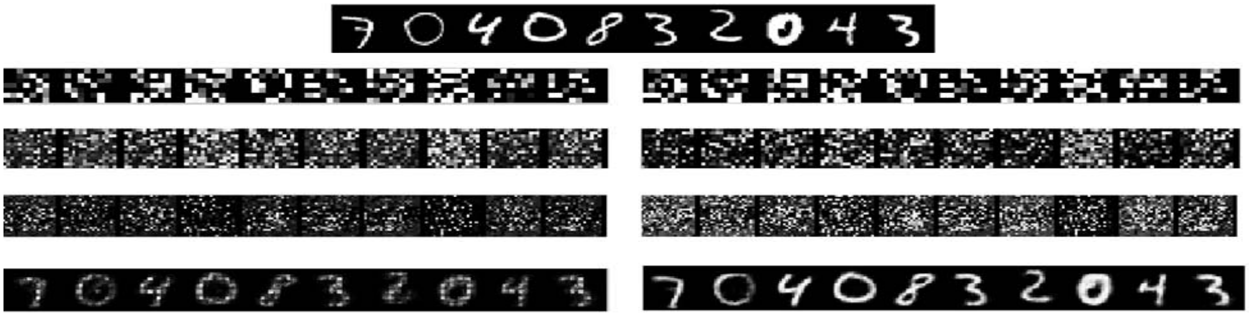


Fig. 3. The reconstructions of LCHN. The top image is the real data. The left column is before BP tuning and the right one is after BPs. Each row represents the outputs of different layers from top to bottom.

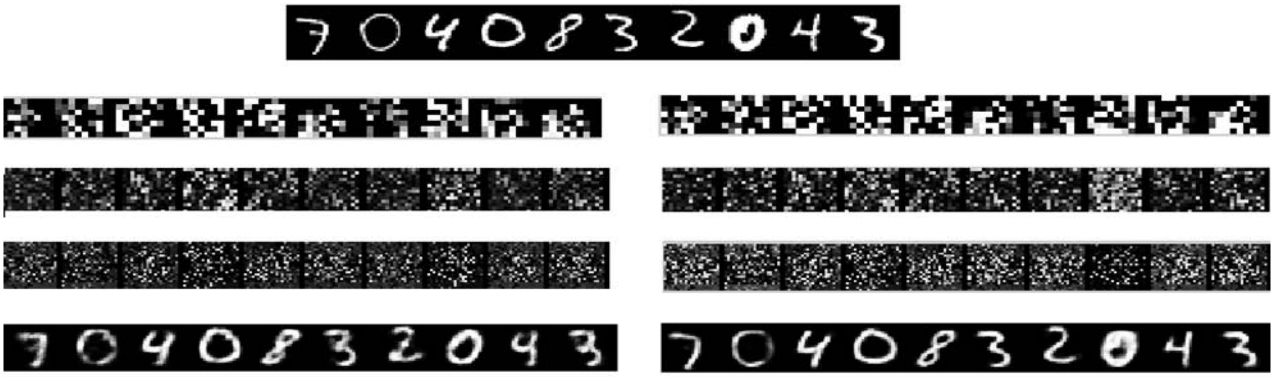


Fig. 4. The reconstructions of LCHN-ID. The top image is the real data. The left column is before BP tuning and the right one is after BPs. Each row represents the outputs of different layers from top to bottom.

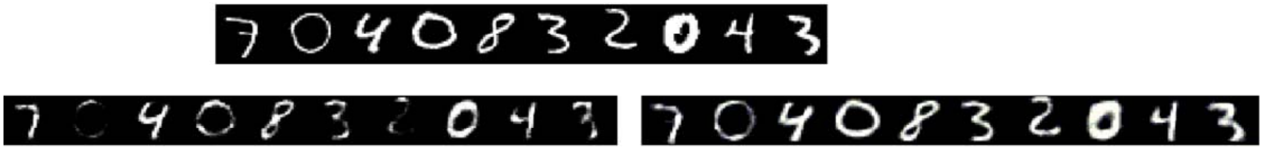


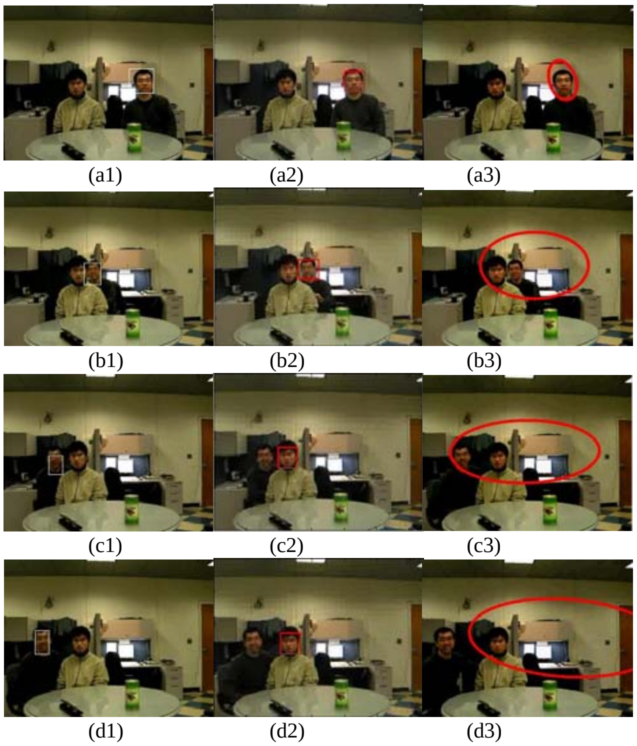
Fig. 5. The reconstructions of fully-connected network. The top image is the real data. The left one is before BP tuning and the right one is after BPs.

Object Tracking using Swarming Particles

Abstract:

This project proposed a new object tracking algorithm that embeds swarming particles into generic particle filter framework to achieve more robustness and flexibility. Firstly a group of particles associated with potential solutions are initialized in a high-dimensional space. Then particle swarm optimization (PSO) is used to drive particles flying. The object is tracked when the particles reach convergence. This PSO-based algorithm contains resample, similarity measure, and integration together such that the degeneracy problem of particle filter can be avoided. Furthermore, a multiple feature model is proposed for object description to enhance the tracking accuracy and efficiency. The proposed algorithm is independent with specific objects and can be used for any free-selected object tracking. Some experimental results demonstrate efficiency and robustness of the algorithm.

Simulation Results:



(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

(d1) (d2) (d3)

Fig.1. An indoor tracking video experiments using PSO-PF, general PF and mean shift methods. First column (a1)(b1)(c1)(d1) shows the results of the proposed PSO-PF method, the second column (a2)(b2)(c2)(d2) for a general PF method, and the last column (a3)(b3)(c3)(d3) for the mean shift method. From (a1) to (d1), the PSO-PF is capable of tracking the object with short occlusion. However the general

PF lost the object when the object is occluded by cluttered backgrounds, as shown in (a2) and (d2). The mean shift lost the object at the very early stage and keeps drifting away in the whole process.

For detailed information on the object tracking part, please refer to the attached paper.

4. Potential Applications

Inspired by the human visual cortex, a local constrained hierarchy network (LCHN) model is proposed to model the object features. One main reason to employ the multiple-layer networks is that this type of models is believed being capable to learn highly complex functions like perception, reasoning, etc., [18]. For LCHN and LCHN-ID, the spatial relations and dependencies of nodes are encoded into the connections between layers and inter-connections among nodes of the same layer. In such a way, the topology of the network is kept and the number of connections is reduced compare with fully-connected networks.

The nodes of the proposed networks can be any type of feature detectors. However, if they have stochastic binary values, the network can be considered as a group of RBMs. Then a greedy learning algorithm can be applied on the network without supervision. This feature would make the learning process to be very convenient with lots of unlabelled data, which would also make the object recognition procedure to be more automatically compared to the supervised learning methods.

However, there is still remaining work need to be conducted to fully understand and employ this hierarchical network. Besides using RBMs, other constraints and learning strategies will be investigated to make this model suitable for more complex problems. In this paper, we only evaluate LCHN approach on a simple object recognition application. More complex and dynamic object recognition applications will be investigated in the future. For example, if the data are sequentially arriving with time-varied patterns, the online learning algorithm is necessary to catch up the pattern variations. Furthermore, the node populations for the layers and the connection topologies should be able to change adaptively according to dynamic environments. In the future, we plan to develop more powerful and efficient hierarchical neural network for object/pattern learning and prediction. The major applications the object/pattern learning and prediction model and PSO-based object tracking algorithm include intruder detection under dynamic environment, security surveillance systems, situation awareness, and urbane search and rescue.

5. Project Assessment

This project has basically met the SOW objectives, although more theoretical improvements need to be conducted and more complex intruder detection case studies need to be conducted in the real world platform. We are working on this part using bio-inspired hierarchical neural network based approach for complex object/pattern recognition right now, and will obtain some promising results in the continuing phase of this project.

The following papers have been published or submitted based on this project.

1. Y. Zheng and Y. Meng, Particle Swarm Optimization based Particle Filter for Free-Selected Object Tracking, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2008)*. Nice, France. Sept 22-26, 2008
2. Y. Zheng and Y. Meng, A local constrained hierarchical network for object appearance learning, 2009 *IEEE International Symposium on Computational Intelligence in Robotics and Automation.* (submitted)
3. Y. Zheng and Y. Meng, A Hybrid Hierarchical Neural Network for Pattern Learning and Prediction, (in preparation)

6. Reference List

[1] P. Viola and M. Jones. Robust real-time face detection. In *ICCV*, vol. 20 (11), pp. 1254-1259, 2001

[2] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. In *PAMI*, volume 23, pages 349–361, 2001.

[3] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, vol2, pp. 264-271, 2005.

[4] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and Bray C. Visual categorization with bags of keypoints. In *Proc. European Conference on Computer Vision*, 2004.

[5] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*. 2006.

[6] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *NIPS*, Vancouver, 2001.

[7] Bekel, H., Bax, I., Heidemann, G., & Ritter, H. Adaptive computer vision: Online learning for object recognition. In *Proc. DAGM-symposium*, pp. 447–453, 2004.

[8] T. M. Cover, and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–7, January 1967.

[9] David Blei, M. Ng, Andrew Y, and Michael Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research. pp. 993-1022, 2003.

[10] G.E. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, vol18, pp. 1527-1554, 2006.

[11] D. Hubel and T. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophys*, vol 28, pp. 229–89, 1965.

[12] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. *CVPR*, vol2, pp. 994-1000, 2005.

[13] Stephan Kirstein, Heiko Wersing, and Edgar Koerner A biologically motivated visual memory architecture for online learning of objects. *Neural Networks.* vol 21, pp. 65—77, 2008.

[14] Björn Ommer and Joachim M. Buhmann, Learning the Compositional Nature of Visual Objects, *CVPR*, 2007.

[15] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition, *CVPR*, 2007.

[16] G. E. Hinton, Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, vol 14, pp. 1771-1800, 2002.

[17] The MNIST database of handwriting digits. http://yann.lecun.com/exdb/mnist/index.html

[18] Yoshua Bengio, and Yann Lecun Scaling learning algorithms towards AI, *Large-Scale Kernel Machines,* MIT Press, 2007.

Attachments

The following are representative of attachments to be included with report sent to the project sponsor. Copies of the following journal/conference papers are attached

1. Y. Zhang and Y. Meng, Dynamic Multi-Robot Task Allocation for Intruder Detection, 2009 *IEEE International Conference on Information and Automation (ICIA 09).* June 22-25, 2009, Zhuhai/Macau, China. (Finalist of best paper awards)
2. Y. Zhang and Y. Meng, A Decentralized Multi-Robot System for Intruder Detection in Security Defense, *2010 IEEE International Conference on Robotics and Automation*. (submitted)

3. Y. Zheng and Y. Meng, Particle Swarm Optimization based Particle Filter for Free-Selected Object Tracking, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2008)*. Nice, France. Sept 22-26, 2008
4. Y. Zheng and Y. Meng, A local constrained hierarchical network for object appearance learning, 2009 *IEEE International Symposium on Computational Intelligence in Robotics and Automation.* (submitted)